bcyly Labs

Triple Match item sorting kit

Setup and Customization Guide

Welcome to Triple Match item sorting kit, a Unity puzzle game project by bcyly Labs. The project includes ready-to-use game screens, item tools, a Level Editor, boosters, and editable reward settings.

This guide shows you how to give the game your own look and create new content using the included tools and Inspector settings. Follow Quick Setup to run the game, then use the chapters to find the feature you want to change.

Website: bcyly Labs Portfolio

Email: bcylylabs@gmail.com

Contents

1. Quick setup . . . 3

2. Find what you want to change . . . 5

3. Rebrand the game . . . 6

4. Add and edit items . . . 10

5. Understand the Level Editor . . . 16

6. Create your first level . . . 21

7. Edit and manage levels . . . 23

8. Generate levels automatically . . . 24

9. Change shelves and obstacles . . . 30

10. Manage boosters . . . 33

11. Change coins, lives, and rewards . . . 39

12. Understand and adjust the game flow . . . 44

13. Change shop products and rewards . . . 49

14. Choose when ads appear . . . 52

15. Change music and sounds . . . 54

16. Important data assets . . . 56

17. Test your changes . . . 57

18. Common questions . . . 58

1. Quick setup

Step 1 - Open the project

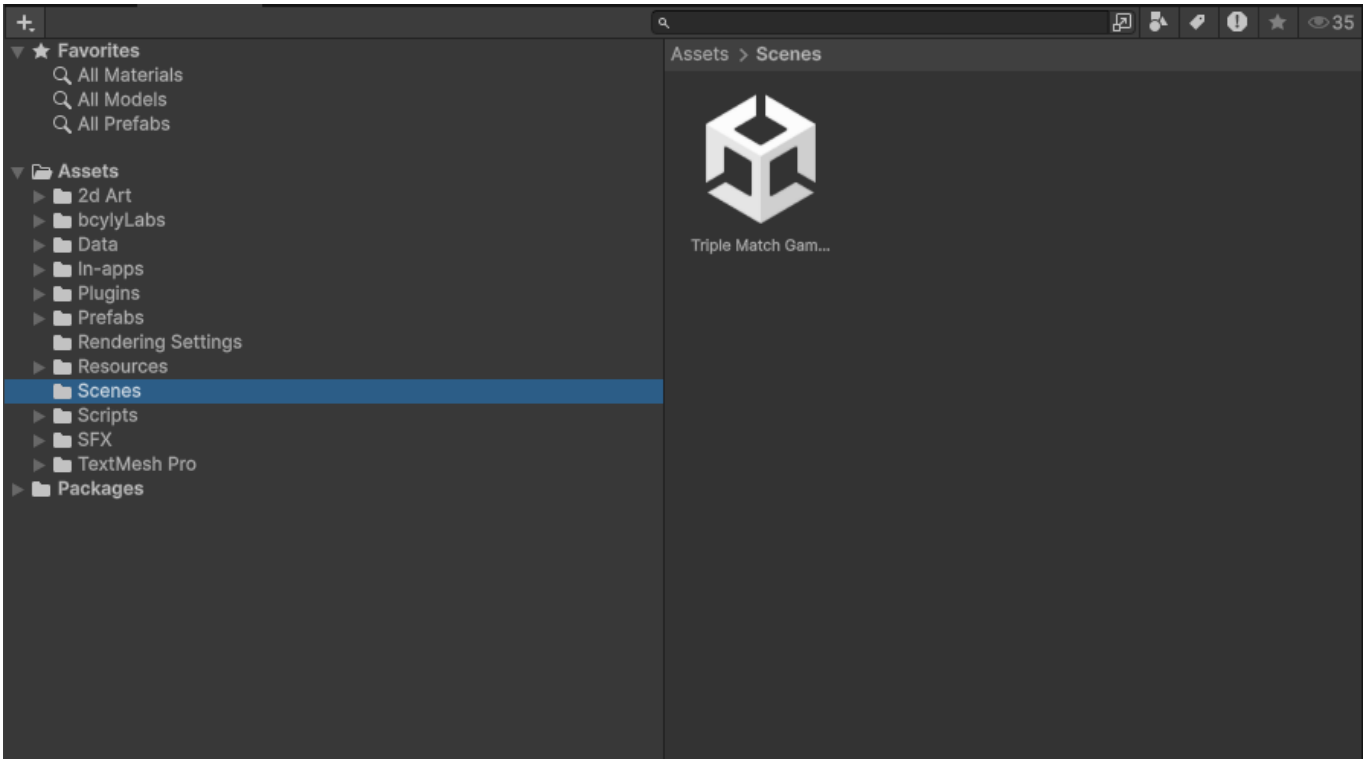
Open Unity Hub. Add the project folder if it is not already listed. Open it with Unity 6000.3.6f1. Wait for Unity to finish importing assets and packages.

Step 2 - Open the main scene

In the Project window, open:

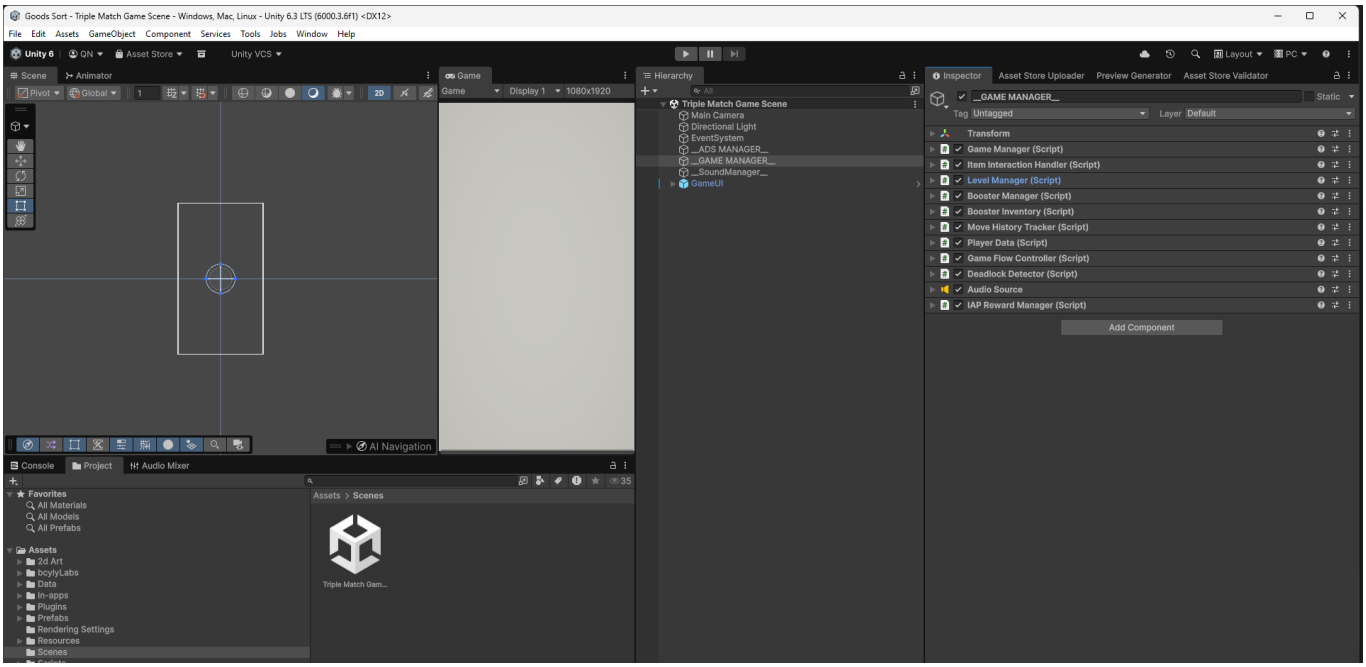
Assets/Scenes/Triple Match Game Scene.unity

Double-click the scene.



Find the main game scene

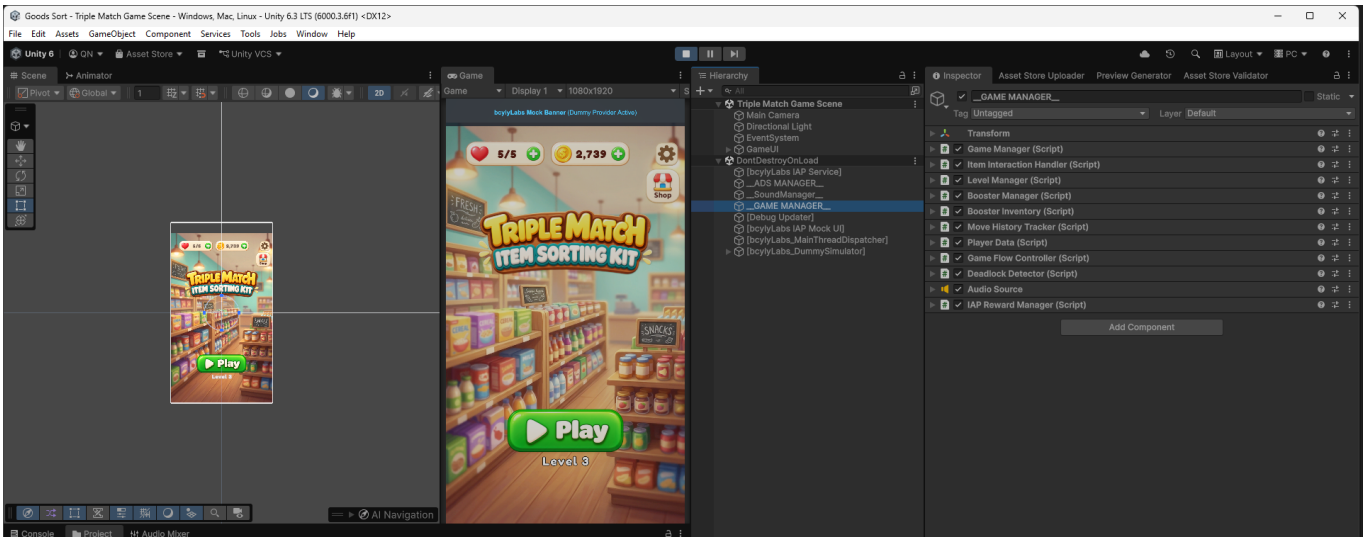
The open scene contains the camera, game managers, and GameUI.



The main scene open in Unity

Step 3 - Play the game

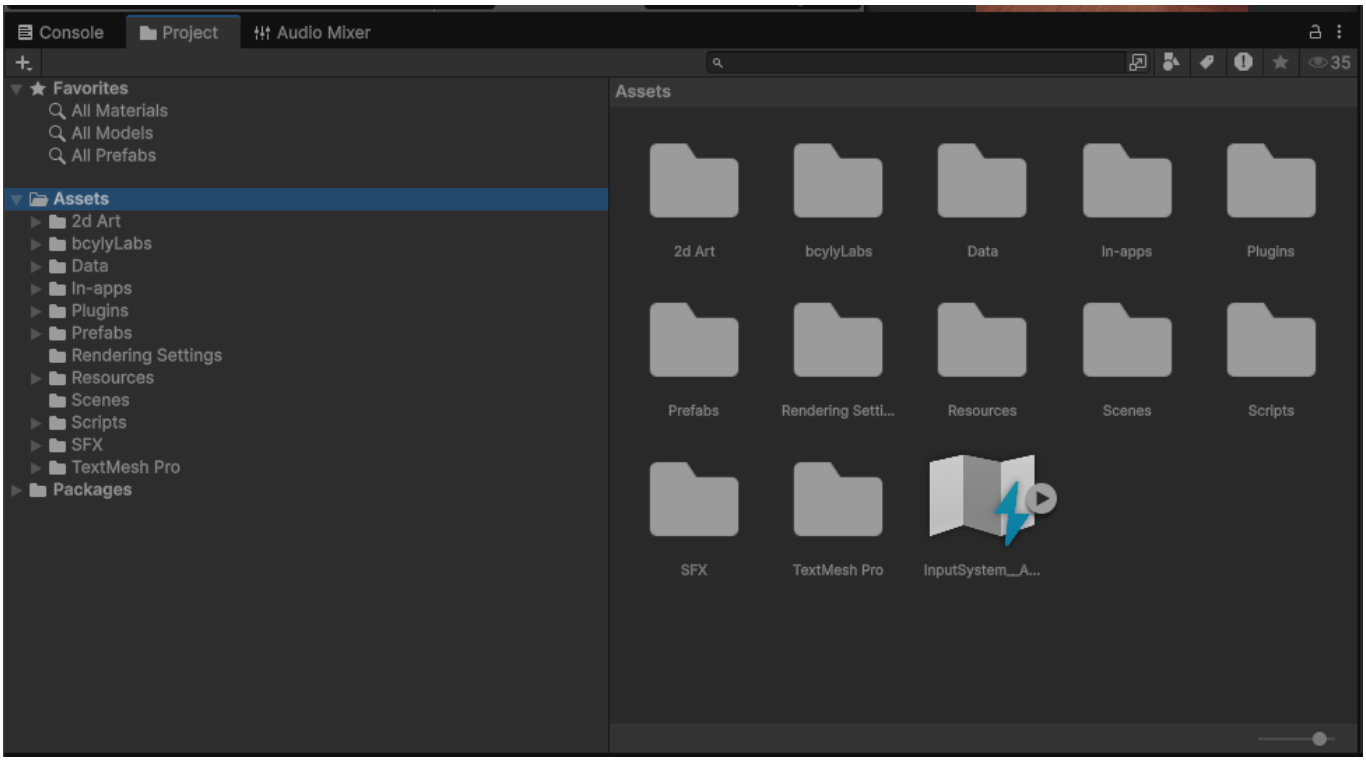
Press Unity's **Play** button. Wait for loading to finish. Press the game's **Play** button and start a level.



Run the game from the main scene

Result: You can open the home screen and start the game. Stop Play Mode before editing content.

2. Find what you want to change



Main project folders

I want to...	Start here
Replace images and fonts	Assets/2d Art and the relevant UI prefab
Add item definitions	Assets/Data/Items
Edit level content	Level Generator & Editor tool
Find saved levels	Assets/Prefabs/Levels
Find the level list	Assets/Data/Levels Database/LevelDatabase.asset
Change shelf graphics	Assets/Prefabs/Shelves
Change common item appearance settings	Assets/Prefabs/ItemPrefab.prefab
Change a screen or popup	Assets/Prefabs/UI Panels
Change booster settings	Assets/Data/Boosters
Change coins and lives	Player Data component on the game manager
Change product rewards	Assets/In-apps/Products

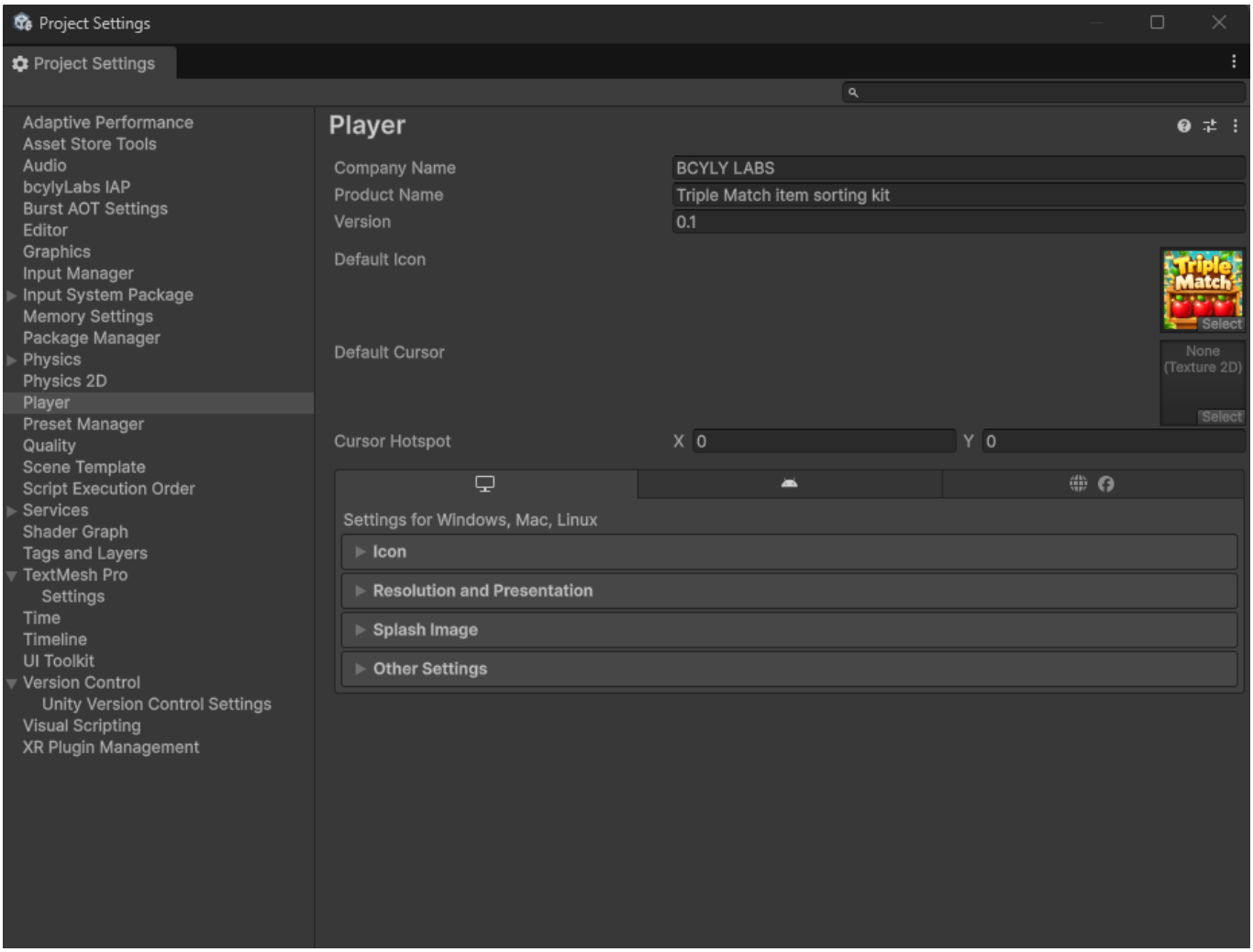
I want to...	Start here
Find the shop product list	Assets/In-apps/IAPDatabase.asset
Change sounds	Assets/SFX/AudioCollection.asset

Tip: Select an assigned field in the Inspector to locate the asset it uses. This is useful when you are unsure which prefab or data asset is active.

3. Rebrand the game

3.1 Change the app name and icon

1. Open **Edit > Project Settings > Player**.
2. Change **Company Name** and **Product Name**.
3. Assign your **Default Icon**.
4. Check the icon settings for your target platform too.



Player settings for the name and icon

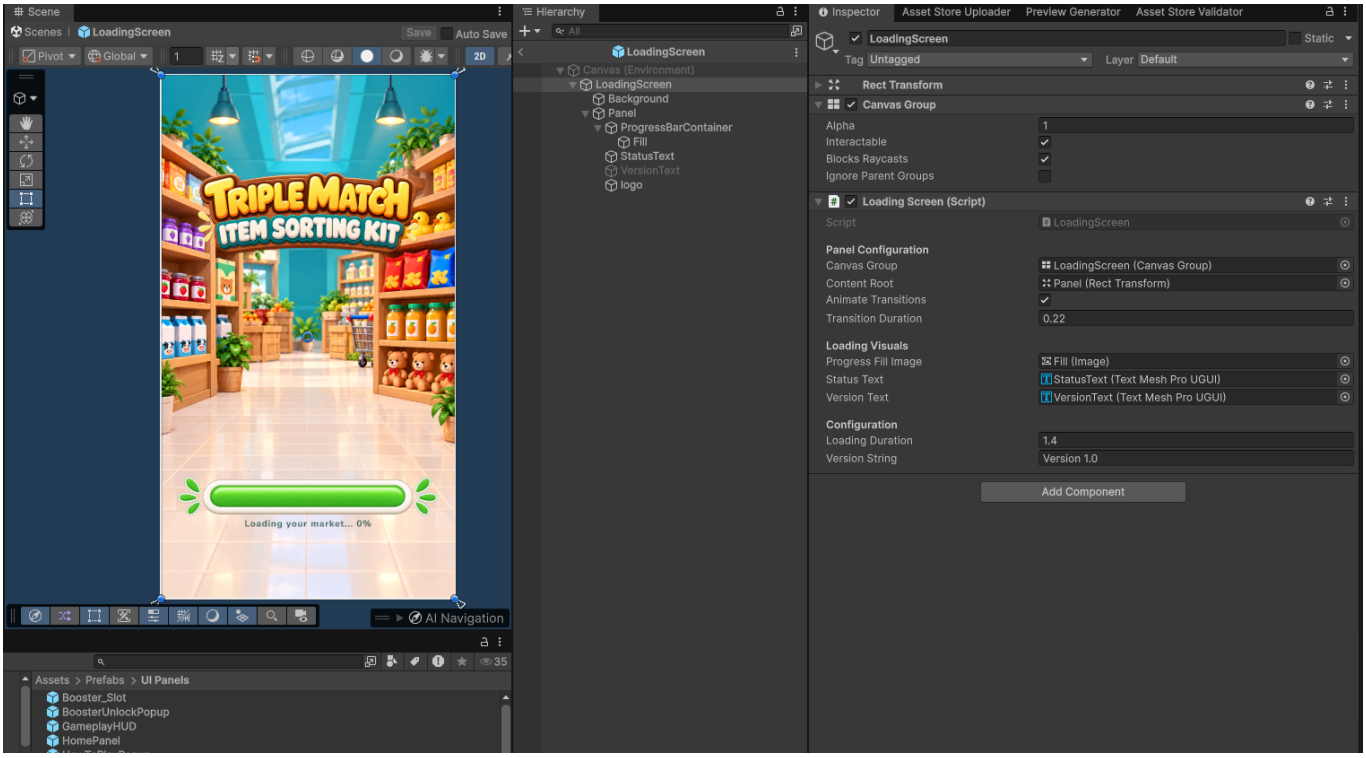
The product name does not replace the logo artwork inside the game. Replace that artwork separately.

3.2 Change the loading screen

Open Assets/Prefabs/UI Panels/LoadingScreen.prefab.

Expand its children in the Hierarchy. Select the background, logo, or text you want to replace. Change the image in its **Image** component, or change the text in its text component.

The Loading Screen component also has references for the progress bar, loading text, and version text. **Loading Duration** changes the displayed loading sequence duration. **Version String** changes its displayed version label.

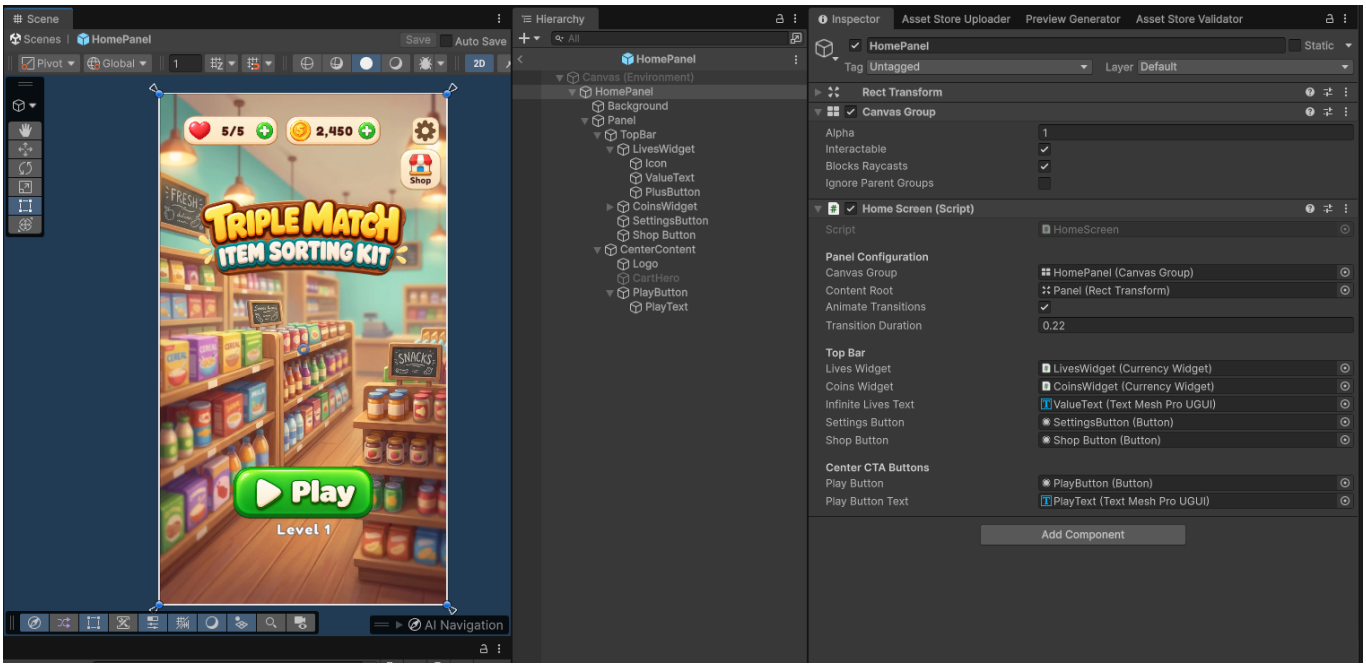


Loading screen layout and settings

3.3 Change the home screen

Open Assets/Prefabs/UI Panels/HomePanel.prefab.

Replace its logo and background. Select the Play button or another UI object to change its graphics. Keep the existing button and text references assigned in the Home Screen component.



Home screen graphics and references

3.4 Change any UI image, button, or text

Use the same steps for the home screen, loading screen, and popups:

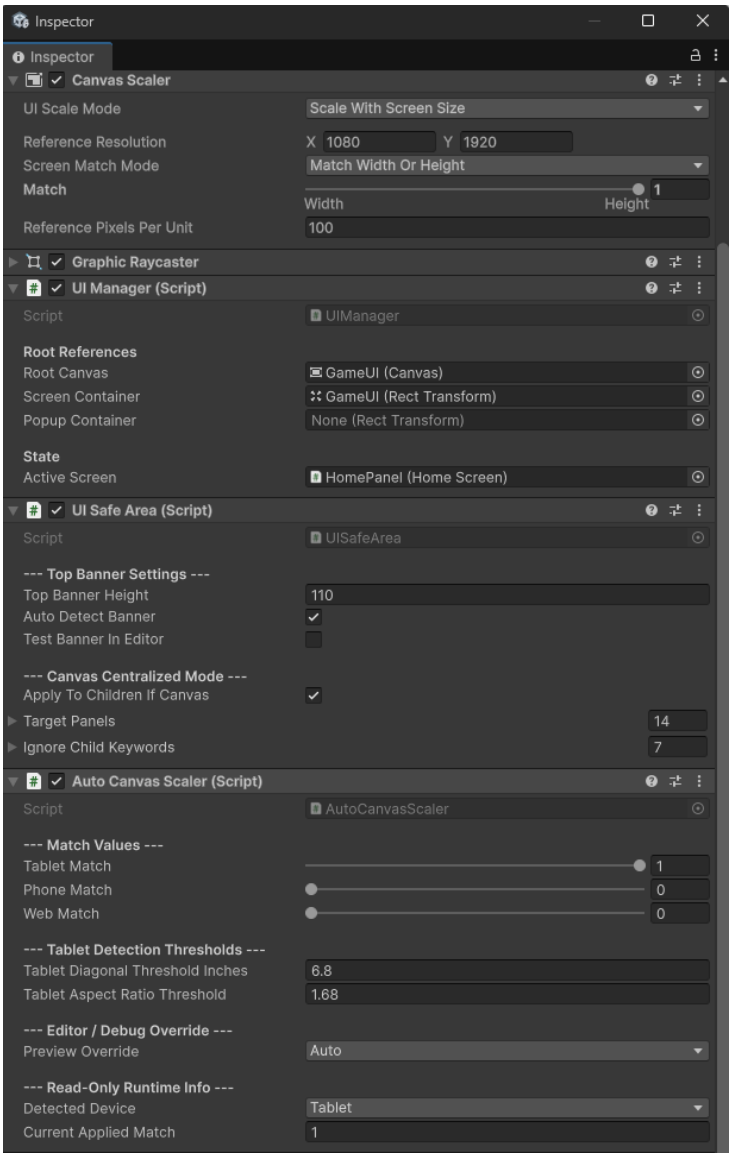
Change	What to select	What to edit
Background or logo	Its child object	**Image > Source Image** and **Color**
Button artwork	The button's graphic object	**Image > Source Image**; check the Button's transition settings too
Text	The text child object	Text, font asset, font size, color, and alignment
Position and size	The UI object	**Rect Transform**
Popup opening animation	The popup root	**Animate Transitions** and **Transition Duration**

Use the loading and home screen images above to find the children. Some text objects use TextMeshPro; some booster labels use Unity's standard Text component. Edit the component that is actually on the object.

For a UI panel with rounded borders, keep the correct sprite borders and Image Type when replacing its sprite. Check the result at its final size.

3.5 Keep the UI readable on different screens

Select the main **GameUI** canvas. Check its Canvas Scaler and Auto Canvas Scaler components.



Canvas and automatic screen scaling settings

Setting	What it does
Reference Resolution	The design size used by the Canvas Scaler.
Tablet Match	Chooses how the UI scales on tablets. The default favors height.
Phone Match	Chooses how the UI scales on phones. The default favors width.
Web Match	Chooses how the UI scales in a browser.
Tablet detection thresholds	Help the game decide whether the screen is a tablet. Usually keep these unchanged.
Preview Override	Lets you preview Auto, Phone, Tablet, or Web behavior.
Runtime information	Shows the detected device and applied match. These are status values.

The automatic scaler can update the Canvas Scaler match value. Use its settings when adjusting device behavior. Return **Preview Override** to **Auto** after testing.

4. Add and edit items

An item needs two things: a sprite and an **Item Data** asset. The sprite gives it a picture. Item Data gives it an identity that the matching system can use.

4.1 Prepare the artwork

1. Import your image into the project, for example into Assets/2d Art/Items.
2. Select the image in the Project window.
3. Set **Texture Type** to **Sprite (2D and UI)**.
4. For one image, use **Sprite Mode: Single**.
5. For a sheet with many items, use **Multiple**, open the Sprite Editor, and slice it.
6. Press **Apply**.

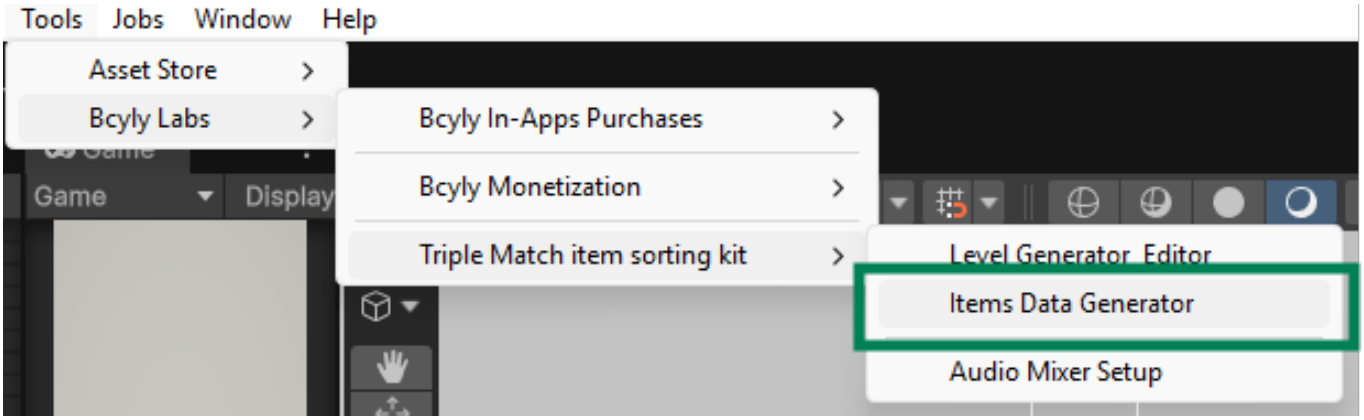
Use transparent backgrounds and similar visible sizes for a consistent set of items. Keep Pixels Per Unit consistent with the existing artwork.



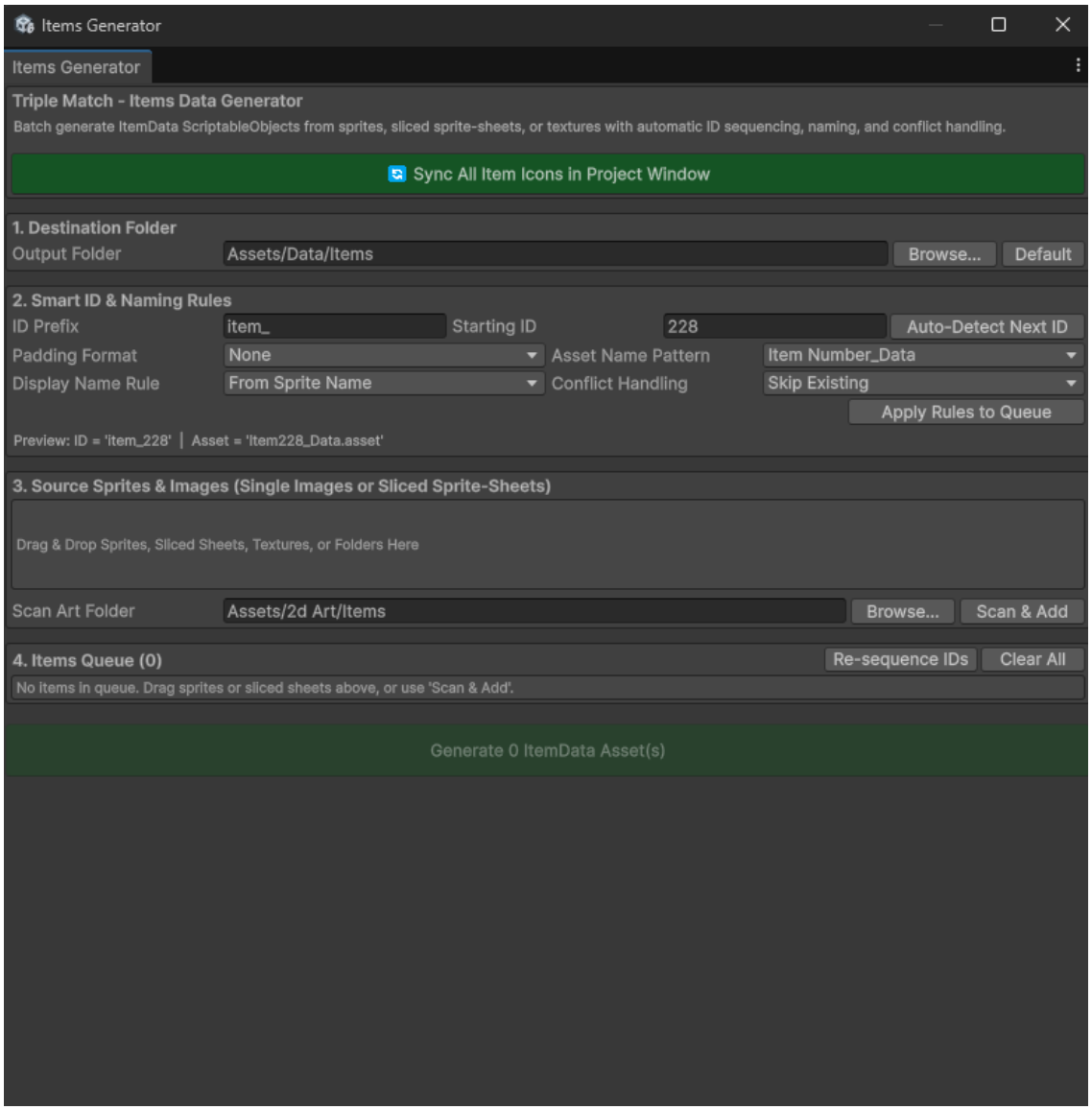
Sprite import settings; the supplied example uses a sheet with many items

4.2 Open the Items Data Generator

Open **Tools > Bcyly Labs > Triple Match item sorting kit > Items Data Generator**.



Open the Items Data Generator



Items Data Generator window

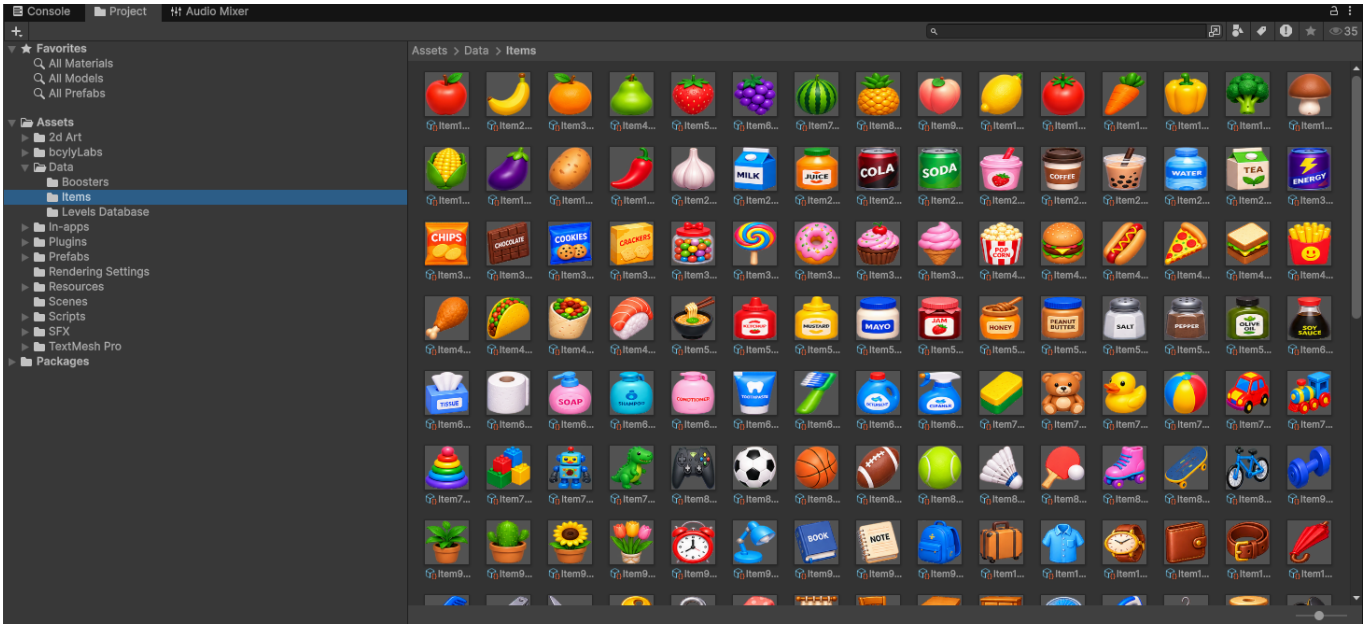
4.3 Generator options

Option	What it does
Sync All Item Icons in Project Window	Refreshes the small preview icons on Item Data assets.
Output Folder	Where new Item Data assets are saved. Use Assets/Data/Items for the normal workflow.
Browse	Selects a folder.
Default	Returns the output path to Assets/Data/Items.
Create Folder	Creates the output folder when it does not exist.
ID Prefix	The first part of each ID, such as item_.
Starting ID	The first number to use for the new items.
Auto-Detect Next ID	Finds the next available number based on the existing assets.
Padding Format	Chooses numbers like 1, 01, or 001.
Asset Name Pattern	Chooses how the saved file is named. Examples include Item1_Data, item_1_Data, a sprite name, a display name, or just the item ID.
Display Name Rule	Uses the sprite name or a numbered name such as Item 1.
Conflict Handling	Chooses what happens when a generated item conflicts with an existing asset or ID.
Apply Rules to Queue	Updates the queued names and IDs using your current rules.
Drag & Drop area	Accepts sprites, textures, sliced sheets, and project folders.
Scan Art Folder / Scan & Add	Finds images in a folder and adds them to the queue.
Convert to Sprites	Appears when queued textures need to be imported as sprites.
Re-sequence IDs	Rebuilds the queue's numbering.
Clear All	Clears the pending queue. It does not delete the existing item assets.
Queue row fields	Let you adjust the asset filename, item ID, and display name before generation.
X on a queue row	Removes that entry from the queue.
Generate ItemData Asset(s)	Creates the queued item definitions.

Conflict choices: Skip Existing leaves the conflicting item unchanged. **Overwrite** replaces existing data. **Auto Increment** looks for another number. For a first attempt, use **Skip Existing** and check the queue messages.

4.4 Create new items

1. Set Output Folder to Assets/Data/Items.
2. Click **Auto-Detect Next ID**.
3. Drag the new sprites into the source area.
4. Check every queue entry. Make sure the IDs are unique.
5. Click **Generate**.
6. Open the output folder and select a new asset.



Item Data assets in the project

4.5 Edit an existing item

Select an Item Data asset in Assets/Data/Items.



An existing Item Data asset and its sprite preview

Field	What it does
Item Id	Identifies the item type for matching. Keep this unique.
Display Name	A readable name for the item.
Icon	The sprite used by the item. Replace this to change its artwork.
Theme Color	Stores the item's theme color. Where it appears depends on the feature using the asset; it does not replace the sprite.
Sprite Preview	Shows the assigned image.
Refresh Project Window Icon	Refreshes this asset's small icon.

To rebrand an existing item: Change its Icon and Display Name. Keep its ID and asset reference when you want existing levels to keep using the same item type.

4.6 Make the item available in a level

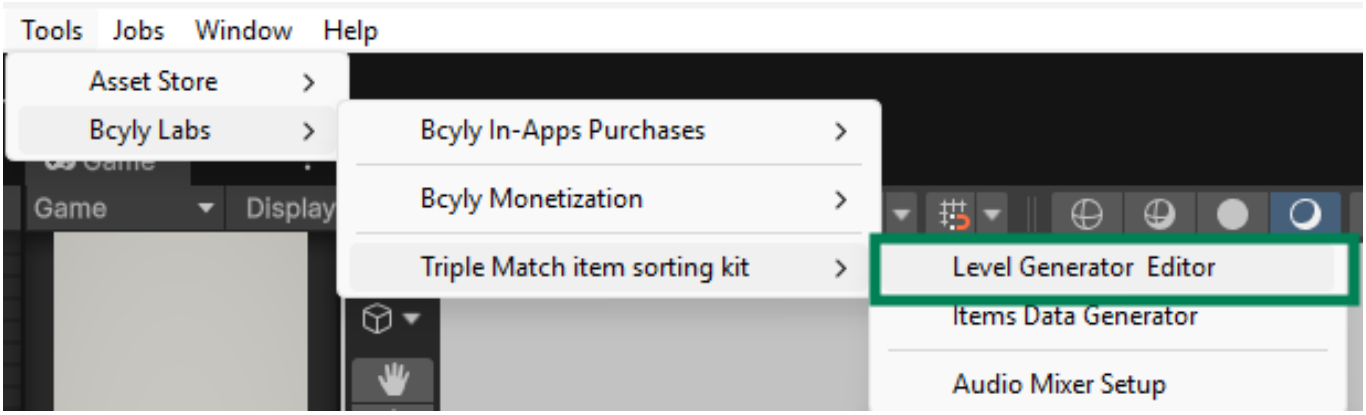
Open the Level Editor and refresh the Item Palette. The tool discovers Item Data under Assets/Data.

- In **Manual** mode, select the new item and paint it into slots.
- In **Auto Distribute** mode, make sure the level's item pool includes it. A saved level can keep its own pool, so creating an asset alone does not guarantee that every old level will use it.
- If **Range** is selected, check that the new item is inside the selected index range.

Save the level and playtest it. See Create your first level.

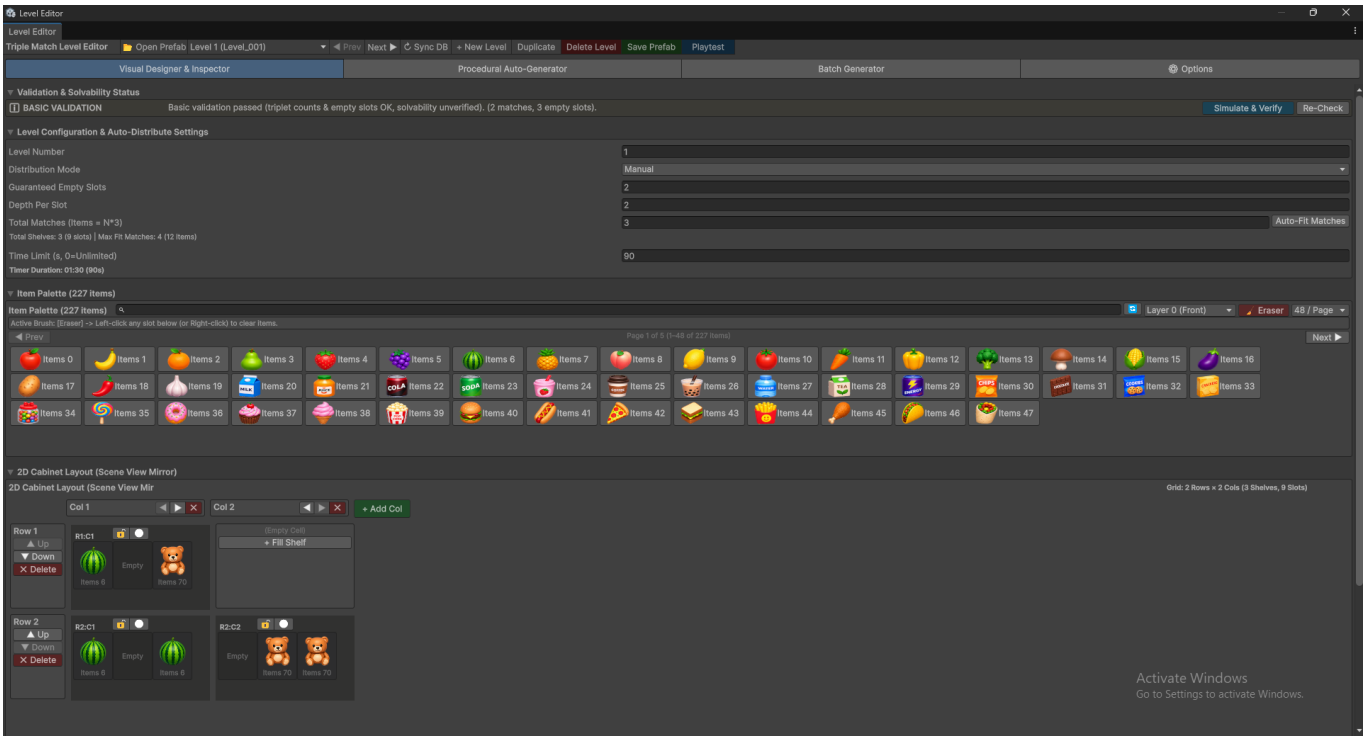
5. Understand the Level Editor

Open **Tools > Bcyly Labs > Triple Match item sorting kit > Level Generator & Editor**.



Open the Level Editor

5.1 The four tabs

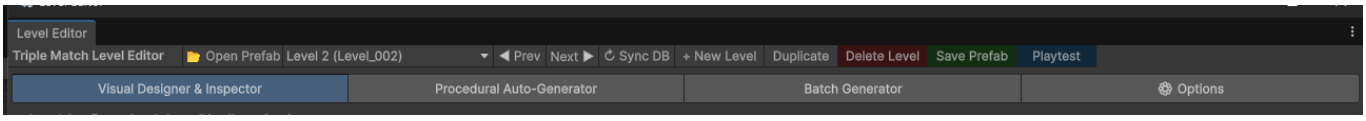


Level Editor window

Tab	Use it for
Visual Designer & Inspector	Build or edit one level by working with shelves and items.
Procedural Auto-Generator	Generate one level from settings such as size, shape, and item count.
Batch Generator	Generate many levels with a difficulty progression.
Options	Generate alphabet/number layouts and run project-wide level maintenance tools.

These tabs are different from **Distribution Mode**, which decides how a level receives its items.

5.2 Top toolbar



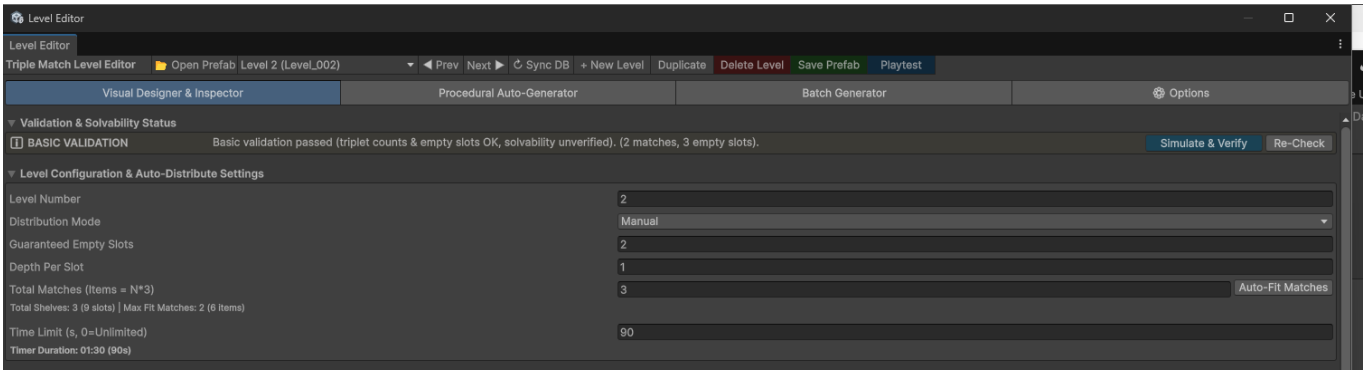
Level selection, saving, and playtest controls

Control	What it does
Open Prefab	Opens the selected level's saved prefab in Prefab Mode.
Level dropdown	Selects a level to edit.
Prev / Next	Loads the previous or next level.
Sync DB	Rebuilds the level database from numbered level prefabs on disk.

Control	What it does
+ New Level	Creates the next level with a starting layout. You can then change it.
Duplicate	Saves the current level, then creates a copy with the next level number.
Delete Level	Deletes the selected level and renumbers the later levels. Read the confirmation before continuing.
Save Prefab	Saves your working level and updates its database entry.
Playtest	Starts a test of the selected level in Play Mode.

Save before changing levels or closing the editor. The visual editor works on an editing copy. Use its Save Prefab button to store your work.

5.3 Level configuration



Visual Designer level configuration

Option	What it means
Level Number	The level's number. Saving uses a name such as Level_001.prefab. Keep it consistent with the level you intend to edit.
Distribution Mode: Manual	Uses the item layout you place in the editor.
Distribution Mode: Auto Distribute	Creates items from the level settings when the level starts.
Use Random Items	Enables random item selection for auto distribution.
Guaranteed Empty Slots	Reserves room for moving items during distribution.
Depth Per Slot	Maximum number of item layers in a slot for distribution. A value of 2 allows a front layer and a layer behind it.
Total Matches	Requested groups of three for generated content. For example, 8 matches means 24 items.
Auto-Fit Matches	Sets the match count to fit the calculated capacity.
Capacity information	Shows shelves, slots, and the estimated maximum item capacity.

Option	What it means
Item Pool Mode: All Items	Uses all items available to the tool.
Item Pool Mode: Range	Uses only the selected item index range.
Start / End Item Index	The first and last item indexes in that range. Check the displayed item names.
Time Limit	Seconds available to finish. 90 means 1 minute 30 seconds. 0 means unlimited time.

In **Manual** mode, your painted items define the layout. Editing Total Matches alone does not paint more items. Use the balance matrix to check the actual placed items.

5.4 Item Palette and painting



Item Palette and cabinet layout

Control	How to use it
Search	Find an item by its name.
Refresh icon	Reload Item Data from disk. Use it after creating items.
Item thumbnail	Select the item as your active brush.
Layer dropdown	Choose Layer 0 (Front), Layer 1 (Back), or Layer 2 (Deep Back).
Eraser	Clears items from clicked slots. Right-click is also a clear action.
Items per page	Changes how many thumbnails appear on one page.
Prev / Next below the palette	Moves between item pages.
Active Brush message	Shows which item and layer you are editing.

Select an item, select the layer, then left-click a slot. Check the front and deeper layers when you edit an existing level.

5.5 Cabinet layout controls

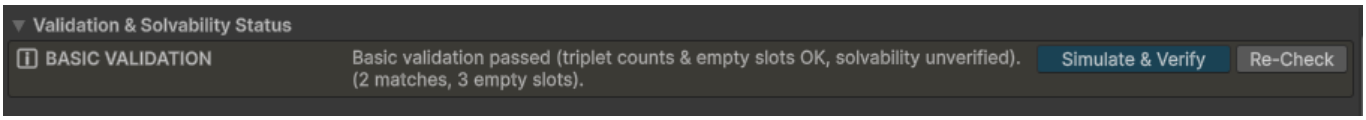
Use the cabinet area below the palette in the same screenshot.

Control	What it does
Add First Row	Starts the layout when no shelves are present.
+ Add Col	Adds another column.
Column arrows	Move a column left or right.
X beside a column	Removes that column.
Row Up / Down	Changes the row order.
Row Delete	Removes the row and its shelves.
+ Fill Shelf	Adds a shelf in an empty grid cell.
+ Add Row (Full Width)	Adds a row across the current width.
Re-align Cabinet Spacing & Scale	Repositions the shelves using the layout settings.
Reshape rows / columns and Apply Reshape	Changes the cabinet arrangement. Check the whole layout afterward.
Horizontal Spacing	Changes the space between columns.
Vertical Spacing	Changes the space between rows.
Reset Spacing (Default)	Restores the tool's spacing values.

Control	What it does
Lock icon on a shelf	Adds or removes the shelf lock.
Required match count near a lock	Sets how many matches are needed to unlock it.
Restricted icon on a shelf	Adds or removes the shelf restriction.
Hold Items	Lets a restricted shelf contain items when enabled.

Small spacing values can make shelves overlap. Use Reset Spacing if you need a clean starting point. The current reset values are about 0.406 horizontally and 0.333 vertically; the middle of the slider is not the exact reset value.

5.6 Check the level



Validation controls

- **Re-Check:** Updates the validation result.
- **Simulate & Verify:** Runs the level solver check.
- **Item Triplet Balance Matrix:** Shows counts for each item type in a manual layout. Counts should form complete groups of three.

For example, 3 apples and 3 bananas form two complete matches. Four apples leave an extra apple.

A validation result is useful, but always play the level too. Check the timer, readability, difficulty, and whether the moves feel clear.

6. Create your first level

Use the Visual Designer and palette screenshots in Section 5 while following these steps.

A small manual level

1. Stop Play Mode and open the Level Editor.
2. Click **+ New Level**. The tool creates a starting layout; it is not an empty scene.
3. Keep the new level number so you do not replace an existing level.
4. Choose **Distribution Mode: Manual**.
5. For a simple test, make a layout with three shelves and three active slots per shelf. Remove extra rows or columns if needed.
6. Set **Depth Per Slot** to 1 and choose **Layer 0 (Front)**.
7. Clear any items you do not want to keep.
8. Choose one item, such as an apple. Place one apple in each shelf's first slot.
9. Choose another item, such as a banana. Place one banana in each shelf's second slot.

10. Leave each shelf's third slot empty.
11. Set **Time Limit** to 90.
12. Use the balance matrix: you should have 3 apples and 3 bananas, with space to move them.
13. Click **Re-Check**, then **Simulate & Verify**.
14. Click **Save Prefab**.
15. Click **Playtest** and complete the level.

Result: You have saved a small level with two item types and two complete matches.

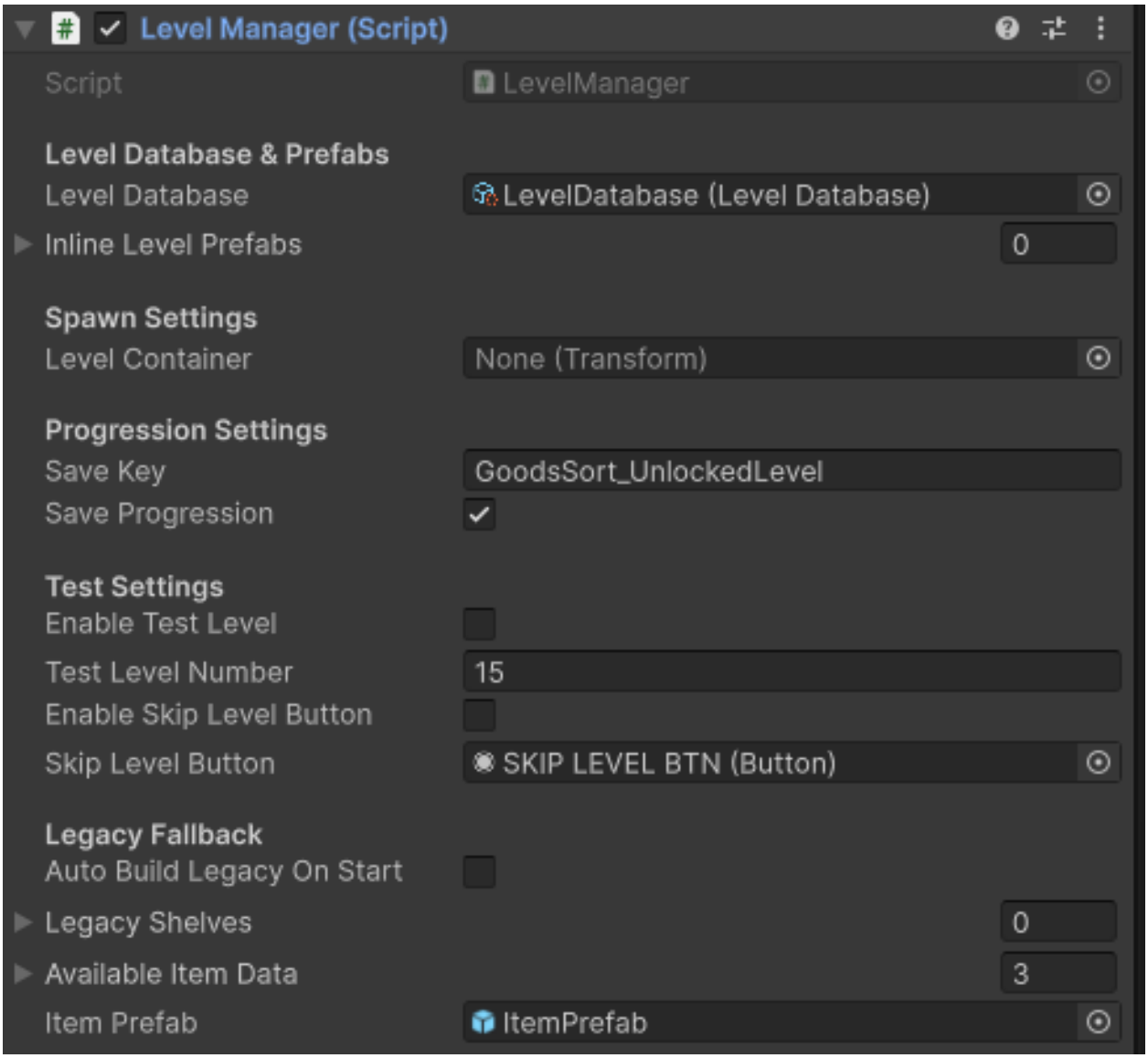
Where is it saved?

The editor saves the level in Assets/Prefabs/Levels and updates Assets/Data/Levels Database/LevelDatabase.asset.

You do not need a separate Unity scene for each level. The game loads level prefabs from its database.

Test a specific level from the main scene

Select the game manager and find **Level Manager**. Turn on **Enable Test Level**, enter its **Test Level Number**, and press Play. Turn test mode off when you finish.



Level Manager testing and database settings

7. Edit and manage levels

Edit an existing level

1. Select it in the Level Editor dropdown.
2. Check its Distribution Mode.
3. In Manual mode, edit the items and their layers.
4. In Auto Distribute mode, edit the pool, match count, depth, and empty space settings.
5. Change shelf layout, modifiers, or time as needed.
6. Run the checks, click **Save Prefab**, and playtest.

Changing from auto to manual is a change in how items are provided. Check and fill the manual layout before saving it as your finished level.

Make a variation

Use **Duplicate** in the toolbar. It saves the source and creates a new numbered copy. Change the copy's items, shape, modifiers, or time. Save it again.

Add or sync level entries

The **Level Database** is the ordered list used by Level Manager. The editor normally updates it for you. **Sync DB** scans numbered prefabs and uses their numbers to rebuild the list.

Keep the Level_001, Level_002, and similar naming pattern. A missing number can leave an empty database entry. Sync does not automatically fill missing level numbers.

Change order or remove a level

For a temporary custom order, you can adjust the database list in the Inspector. However, **Sync DB restores the order from prefab numbers**. Keep prefab names, Level Number values, and database order consistent for a lasting change. Back up levels before renumbering or swapping them.

Delete Level removes the selected level and shifts later numbers down. It affects more than the visible level. Check the new order afterward, especially if a tutorial or unlock is tied to a level number.

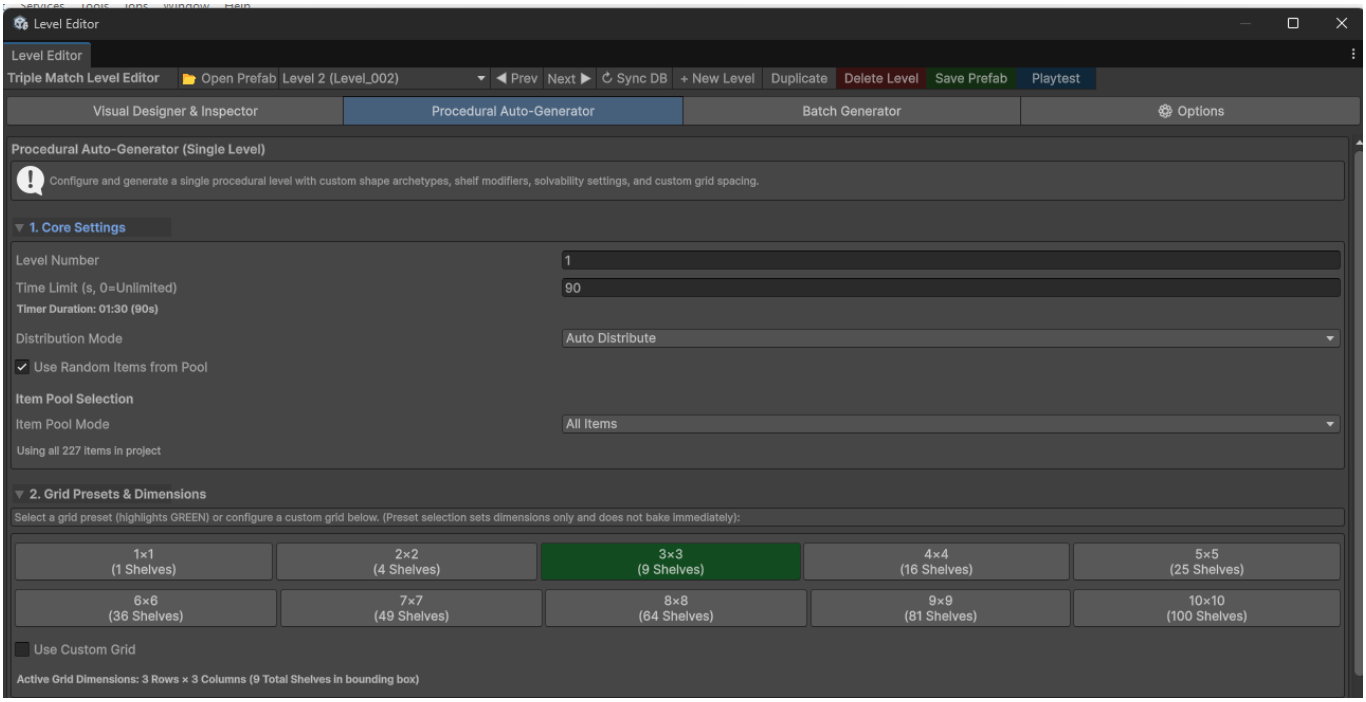
Scene copy or saved prefab?

Edit the saved level through the Level Editor for regular content work. Changing a temporary running level in Play Mode does not update its saved prefab.

8. Generate levels automatically

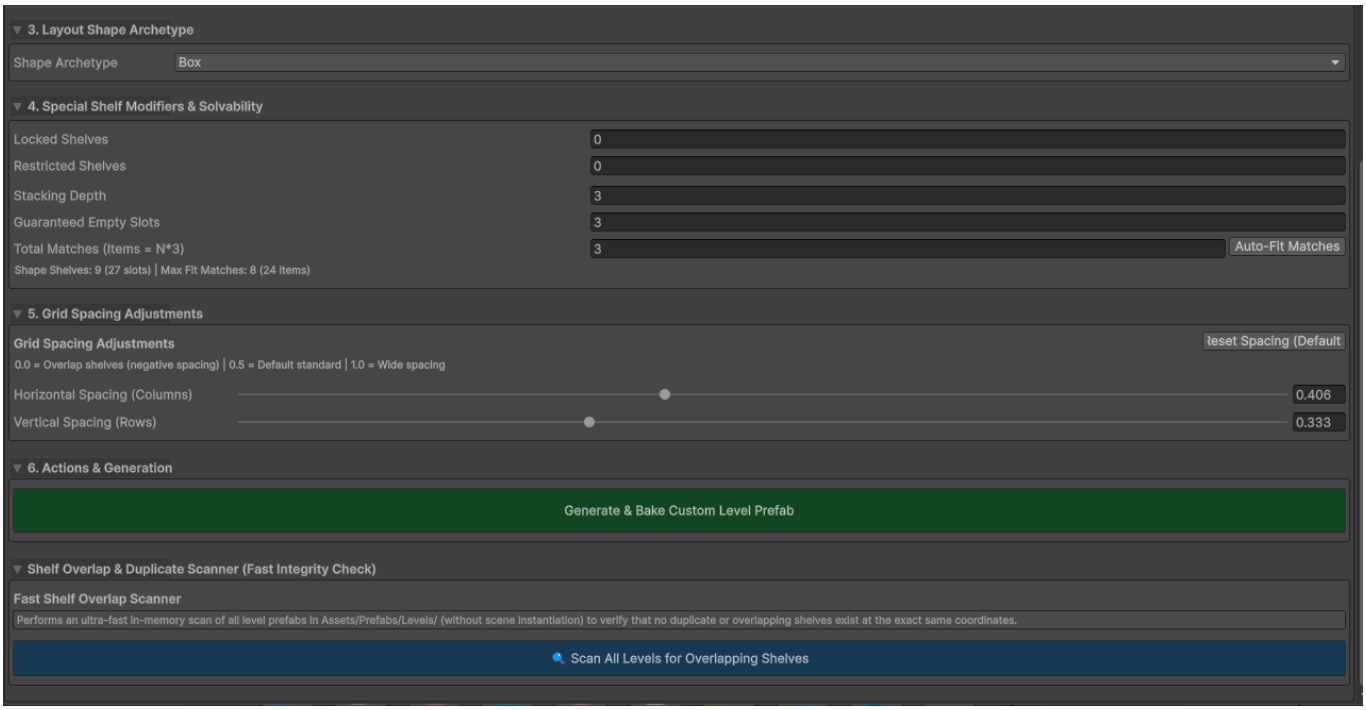
8.1 Generate one level

Open **Procedural Auto-Generator**.



Single-level generator: core settings and grid

Setting	What it does
Level Number	Chooses the numbered level to generate. Use an unused number for new content.
Time Limit	Sets the timer; 0 means unlimited.
Distribution Mode	Chooses saved/manual placement or runtime auto distribution.
Use Random Items from Pool	Uses random choices from the selected pool.
Item Pool Mode / Range	Chooses all items or a limited index range.
Grid preset	Chooses dimensions from 1 x 1 to 10 x 10. Selecting a preset does not generate immediately.
Use Custom Grid	Lets you set rows and columns separately, from 1 to 10.
Active Grid Dimensions	Shows the bounding grid. The chosen shape may use fewer shelves.



Single-level generator: shape, modifiers, spacing, and generation

Setting	What it does
Shape Archetype	Chooses the shelf arrangement.
Alphabet Character	Chooses a letter when the Alphabet shape is active.
Notches Count	Chooses how many edge cutouts to use with Random Notches.
Locked Shelves	Number of locked shelves to add.
Matches Required to Unlock	Matches needed before those locks open.
Restricted Shelves	Number of restricted shelves to add.
Restricted Shelves Can Contain Items	Allows items inside those shelves.
Stacking Depth	Maximum layers per slot.
Guaranteed Empty Slots	Reserves room for moves.
Total Matches / Auto-Fit Matches	Sets the requested groups of three or fits the capacity.
Horizontal / Vertical Spacing	Changes the gaps between shelves.
Generate & Bake Custom Level Prefab	Creates and saves the configured level.
Shelf overlap scanner	Checks saved levels for duplicate or overlapping shelves.

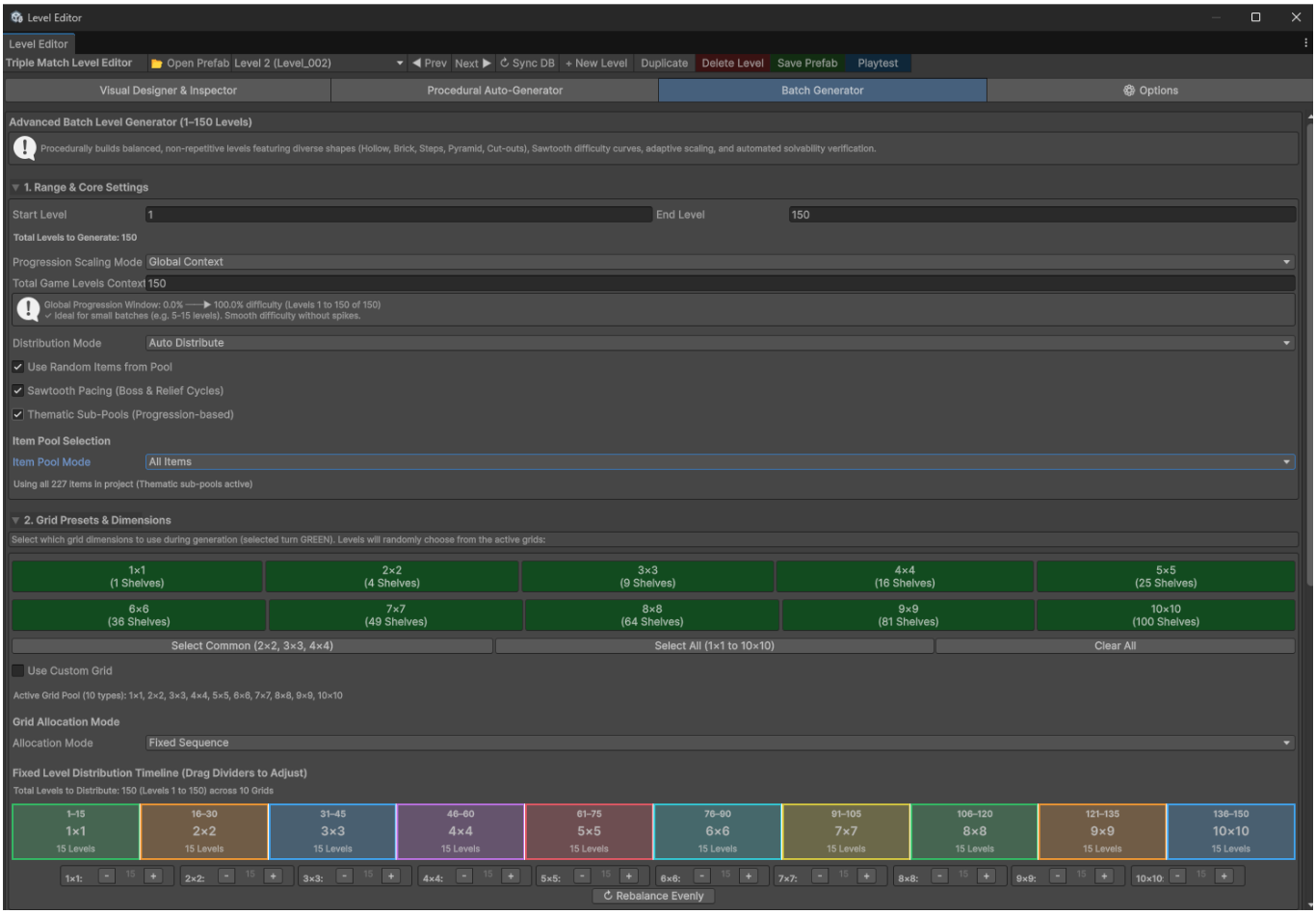
Shape choices: Box, Hollow Ring, Staggered Brick, Staircase, Inverted Staircase, Pyramid, Inverted Pyramid, Left Pyramid, Right Pyramid, Cut-Out Center, Random Notches, Alphabet, and Cross/Plus.

Easy starting recipe: Use an unused level number, a 3 x 3 grid, Box shape, Auto Distribute, no modifiers, depth 1, several empty slots, and 90 seconds. Generate it, open it in the Visual Designer, check it, and playtest.

Generating into an existing level number can replace its content. Check the number before generating.

8.2 Generate a batch

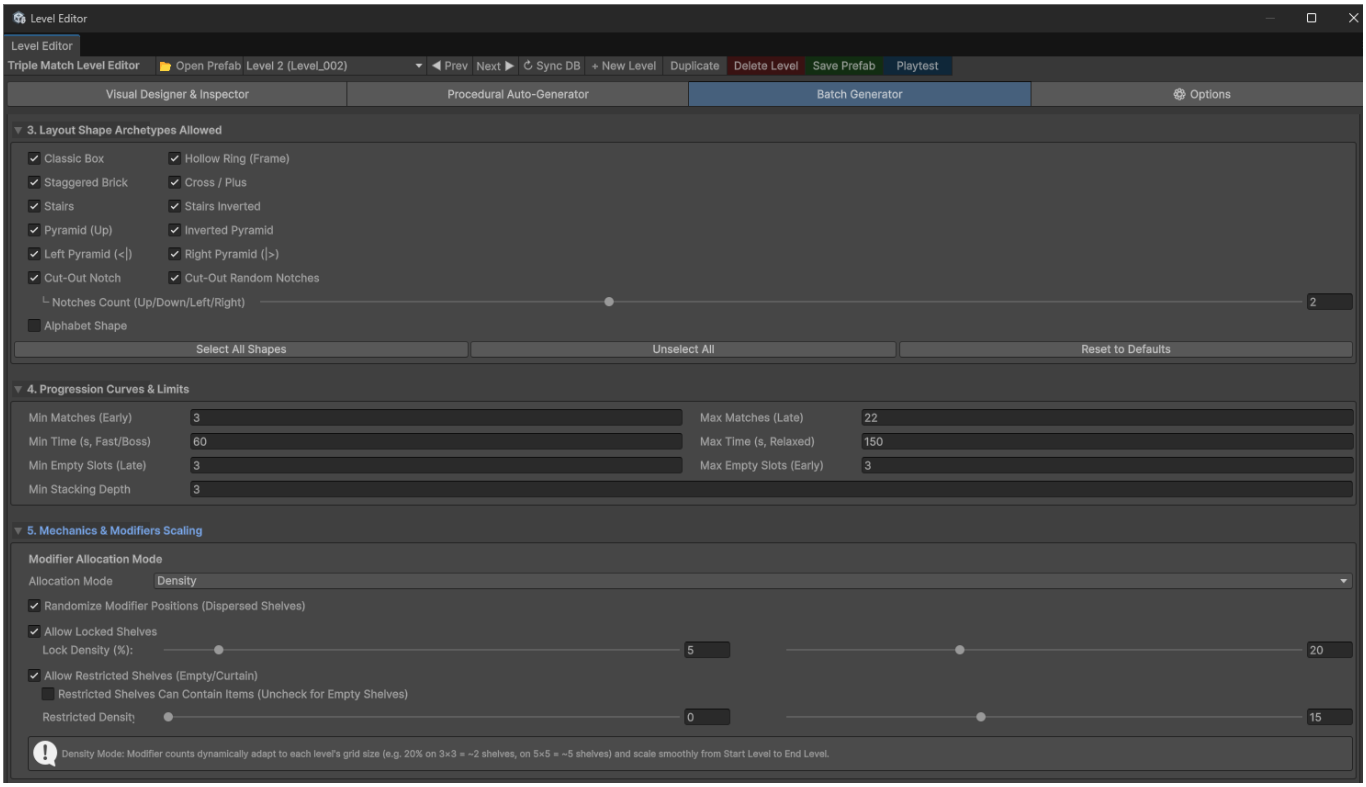
Open **Batch Generator**. Start with a small unused range so you can check the results before making a large set.



Batch generator: range, progression, items, and grid settings

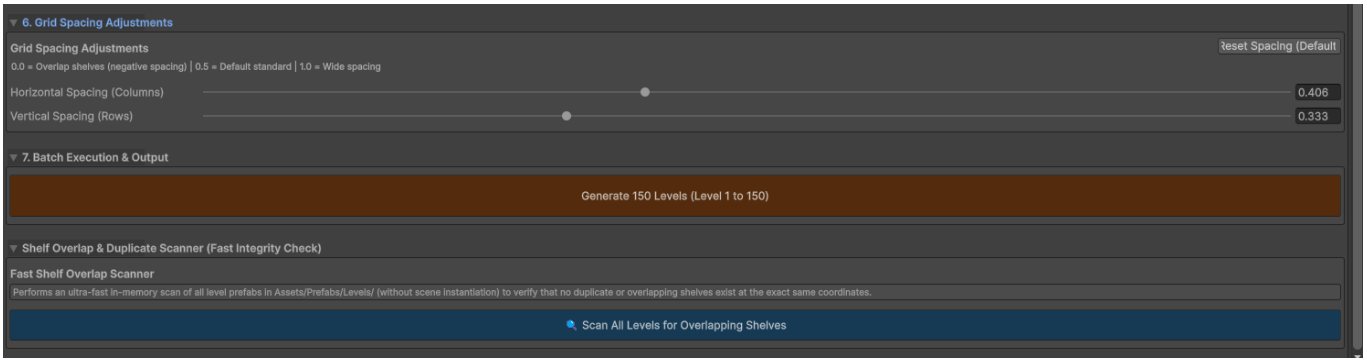
Setting	What it does
Start Level / End Level	The inclusive range to create. For example, 20-24 creates five levels.
Progression Scaling Mode	Chooses how difficulty changes across the range.
Global Context	Uses each level's position in the whole planned game. Useful for adding a small batch to a longer game.
Total Game Levels Context	The total planned level count for Global Context.
Batch Relative	Runs from easy to hard inside this batch.
Custom Range	Uses your selected start and end difficulty.
Start / End Difficulty	Values from 0 to 1. A value of 0.3 means 30% of the generator's difficulty scale.

Setting	What it does
Distribution Mode	Chooses Manual or Auto Distribute.
Use Random Items from Pool	Enables random item choices.
Sawtooth Pacing	Adds harder and easier cycles instead of making every level harder than the last.
Thematic Sub-Pools	Uses smaller item groups as progression changes.
Item Pool Mode	Chooses All Items or Range. A specific Range bypasses the thematic sub-pool choice.
Start / End Item Index	Limits the usable items.
Grid buttons	Enable the grid sizes the batch may use. Green means selected.
Select Common / Select All / Clear All	Quickly changes the selected grid set.
Use Custom Grid	Adds custom row and column dimensions.
Allocation Mode: Random	Chooses grids from the active set.
Allocation Mode: Fixed Sequence	Assigns a planned number of levels to each selected grid size.
Fixed distribution timeline	Drag its dividers, use count controls, or Rebalance Evenly to change how many levels use each grid.



Batch generator: shapes, difficulty limits, and shelf modifiers

Setting	What it does
Shape toggles	Choose which shape types are allowed.
Notches Count / Alphabet Character	Configure the relevant shape.
Select All Shapes / Unselect All / Reset to Defaults	Quickly changes the allowed shapes.
Min Matches (Early) / Max Matches (Late)	Controls the early-to-late item workload.
Min Time (Fast/Boss) / Max Time (Relaxed)	Sets the available timer range.
Min Empty Slots (Late) / Max Empty Slots (Early)	Gives later levels less movement room and early levels more room.
Min Stacking Depth	Sets the minimum starting depth for generation.
Modifier Allocation Mode: Density	Calculates modifier counts as a percentage of shelves.
Modifier Allocation Mode: Range	Uses a minimum and maximum shelf count.
Randomize Modifier Positions	Spreads modifiers across different shelves.
Allow Locked Shelves	Includes locks using the density or count range.
Allow Restricted Shelves	Includes restricted shelves using the density or count range.
Restricted Shelves Can Contain Items	Lets generated restricted shelves hold items.

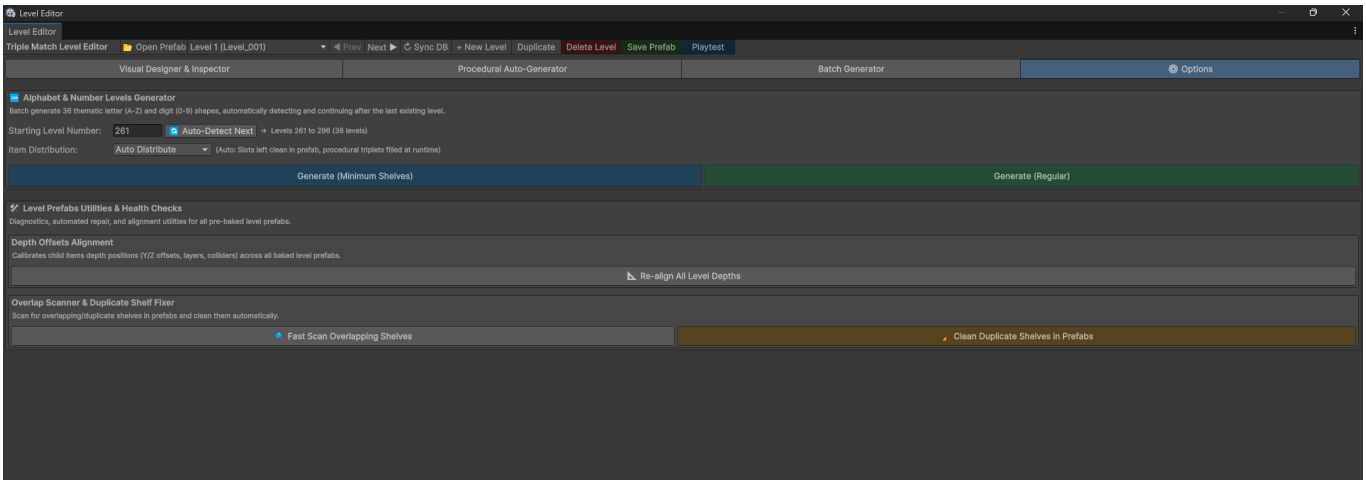


Batch generator: spacing, generation, and overlap scan

Set spacing, check the level range, then press **Generate**. After generation, inspect a level near the start, middle, and end of the batch. Run checks and playtest them.

Generation uses several settings together. More items, more depth, fewer empty spaces, and more locks can all increase difficulty. Change a few settings at a time.

8.3 Options tab



Alphabet/number generation and maintenance tools

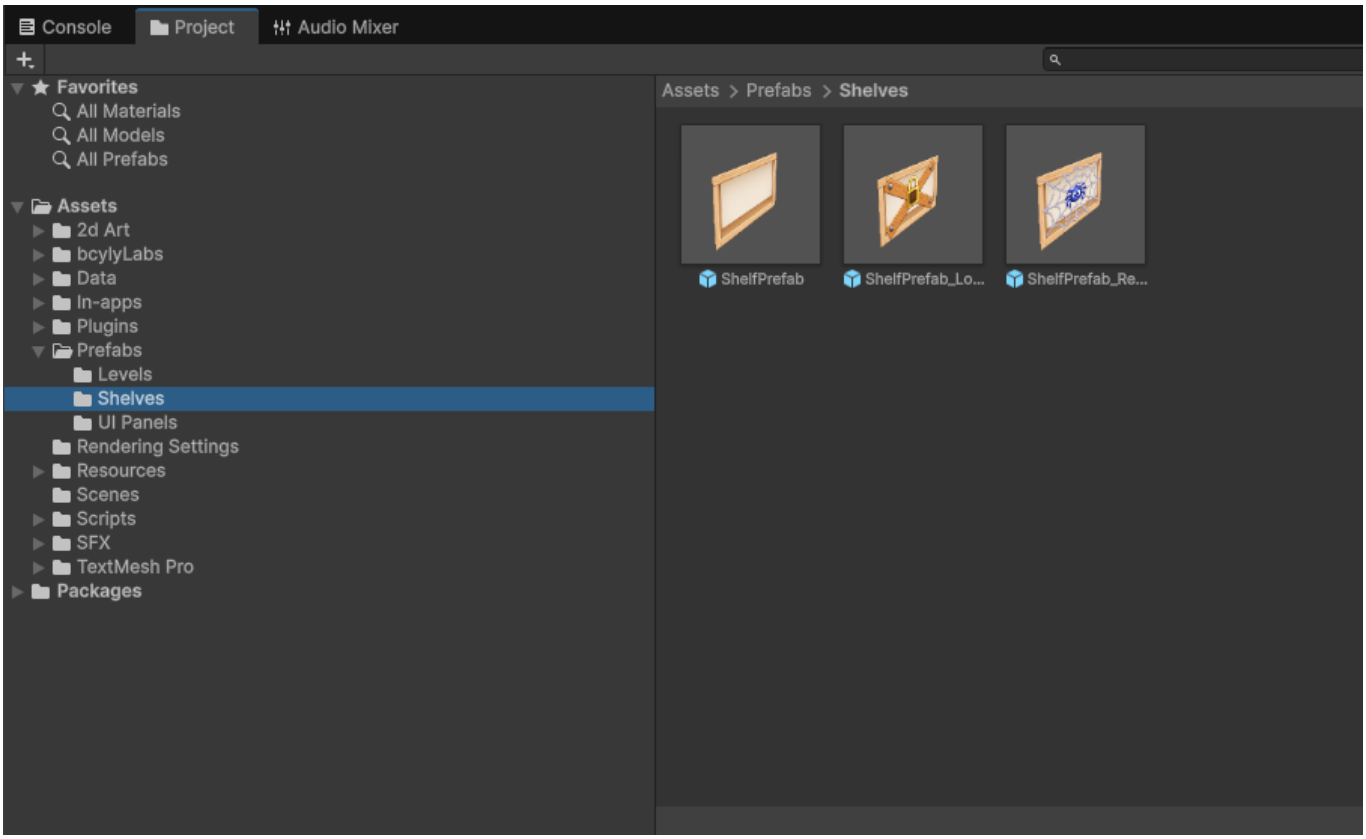
Option	What it does
Starting Level Number	First level number for the alphabet/number batch.
Auto-Detect Next	Starts after the last detected level.
Item Distribution	Chooses Manual or Auto Distribute.
Generate (Minimum Shelves)	Creates compact letter and number layouts.
Generate (Regular)	Uses the regular 5 x 5 bitmap layout style.
Re-align All Level Depths	Updates saved item depth positions and related setup across baked levels.
Fast Scan Overlapping Shelves	Reports duplicate or overlapping shelf positions.
Clean Duplicate Shelves in Prefabs	Removes duplicate shelves from saved level prefabs.

The alphabet/number batch creates **36 layouts: A-Z and 0-9**. Maintenance tools can change many saved levels. Use them when needed and keep a backup first.

9. Change shelves and obstacles

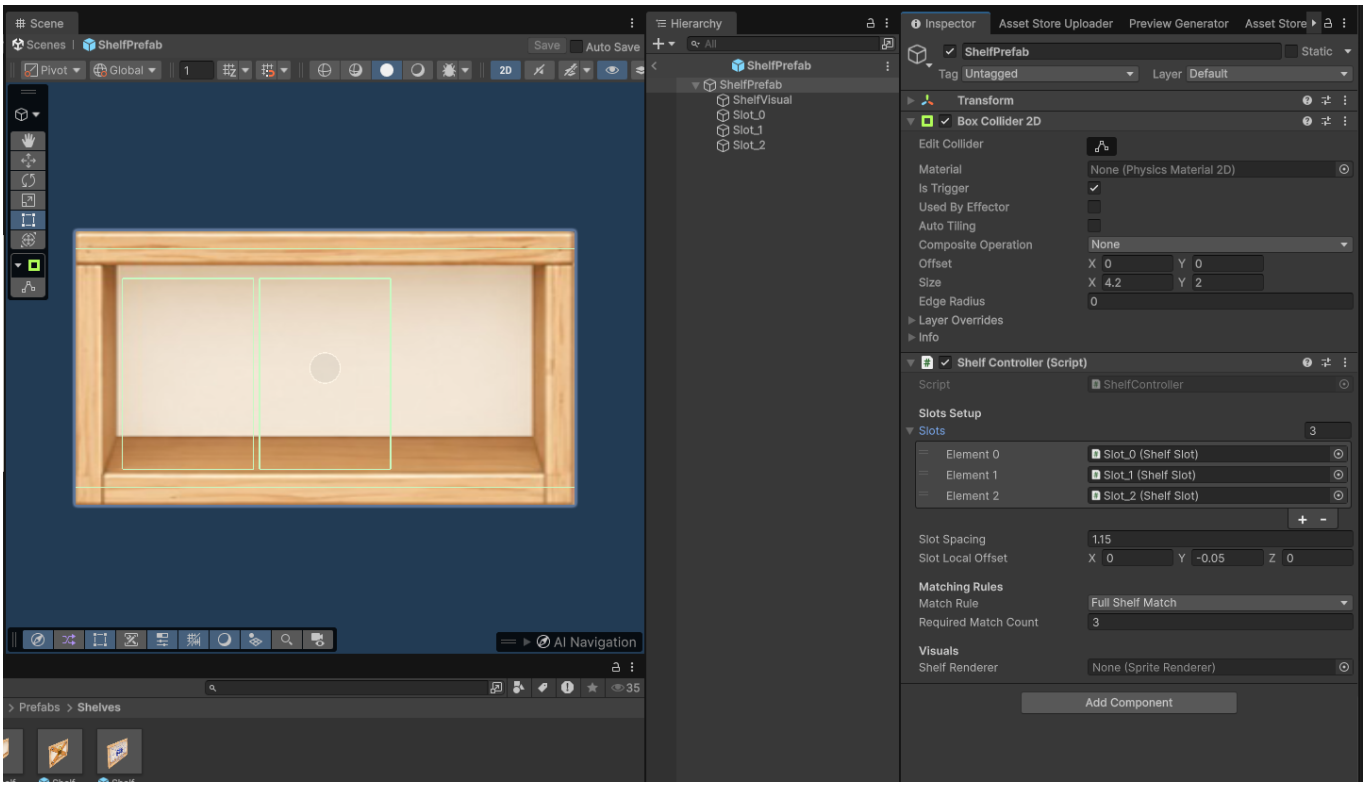
9.1 Replace shelf graphics

Open Assets/Prefabs/Shelves.



Normal, locked, and restricted shelf prefabs

1. Open ShelfPrefab.prefab.
2. Select the object with the shelf's Sprite Renderer. In the supplied example, this is the shelf root.
3. Assign your new image to **Sprite Renderer > Sprite**.
4. Keep its size and pivot close to the original while you test.
5. Save, then check it in an existing level and a newly generated level.
6. Check ShelfPrefab_Locked and ShelfPrefab_Restricted too. Replace their artwork or overrides as needed.



Shelf hierarchy, Sprite Renderer, and Shelf Controller

Changing the source prefab updates linked settings. A level with an overridden sprite may keep its own image. Open that level and check its overrides if it does not change.

9.2 Shelf settings

Field	What it does
Slots	References the shelf's item positions. Keep these assigned.
Slot Spacing	Changes horizontal distance between slots.
Slot Local Offset	Moves the slot arrangement inside the shelf.
Match Rule: Full Shelf Match	Requires the active shelf slots to be filled with matching items.
Match Rule: Triple Match Anywhere	Resolves groups of three matching items on that shelf.
Required Match Count	Sets the count used by the matching rule. Keep 3 for the standard game.
Shelf Renderer	References the Sprite Renderer used by the shelf.

The standard game counts a completed match as three removed items. Keep the standard three-item rules for the documented no-code workflow; changing the count alone is not a complete new matching system.

Select a slot child to find **Slot State** and **Back Offset**. Slot State controls its availability. Back Offset positions deeper items relative to the front. Adjust these only when you need to change shelf behavior or item alignment.

9.3 Item appearance on shelves

Open Assets/Prefabs/ItemPrefab.prefab to adjust shared Item View settings:

- **Front Scale / Back Scale:** Size of front and deeper items.
- **Back Dim Color:** Tint used for items behind the front.
- **Auto Fit To Slot / Target Slot Bounds:** Fits different sprites into a common display area.
- **Base Front Sorting Order / Order Step Per Layer:** Controls which item draws in front.
- **Progressive Scale Step:** Changes scale as depth increases.

Change one setting at a time and check a level with more than one item layer.

9.4 Shelf modifiers

Use the lock and restriction controls in the Level Editor for common setup. For additional settings, select the shelf inside its prefab and expand the relevant component.

Feature	Editable settings and meaning
Shelf Lock	Is Locked, unlock condition, Matches Required, and Key Item. Visual references connect the lock image and counter.
Shelf Restricted	Allow Placement, Allow Pick Up, Can Contain Items, and the restriction visual/tint. These let you control which moves are allowed.
Shelf Mover	Movement Type, Move Delta, Cycle Duration, Is Moving, and Pause During Match. Move Delta is the travel distance and direction.
Slot Obstacle	Clear On Shelf Match and references to its visual. It blocks the slot until cleared.

A modifier's image is separate from its behavior. Replace the lock, restriction, or obstacle visual without removing the component that controls it.

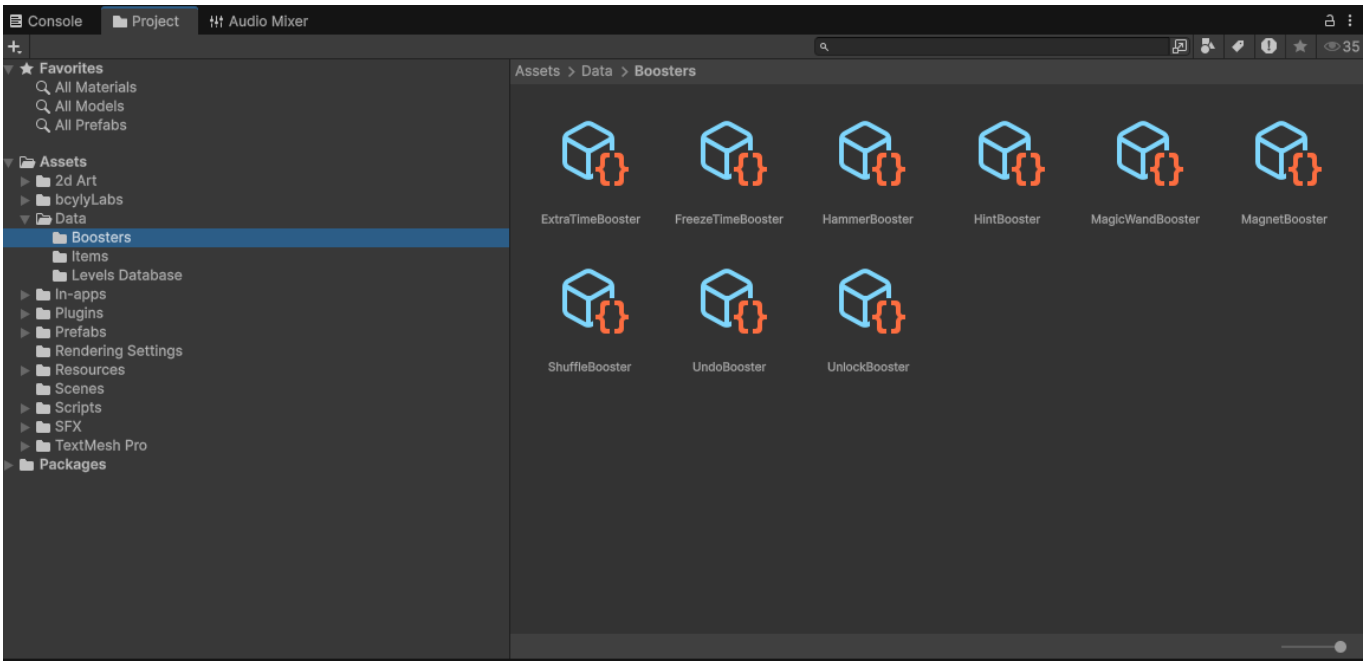
Lock conditions: Match Count Threshold opens after the required number of matches anywhere in the level. Key Item Matched opens when the chosen item type is matched. Manual waits for an existing external action, such as Unlock.

Movement types: Ping Pong travels back and forth between the starting point and Move Delta. Sine Wave oscillates around the starting point. Give moving shelves enough space so their paths stay readable.

10. Manage boosters

10.1 Find the booster assets

Open Assets/Data/Boosters.



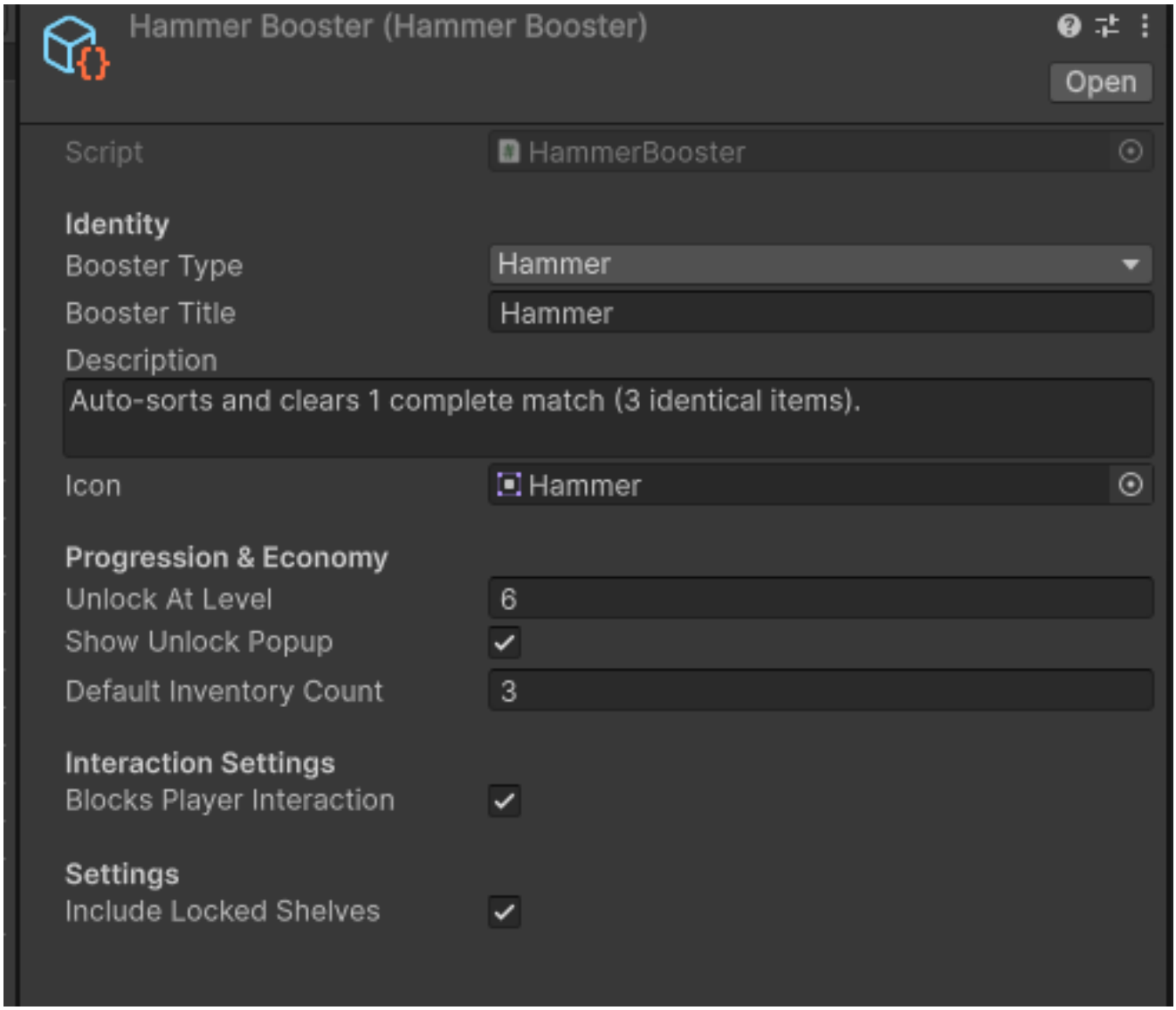
Available booster assets

Booster	What it does	Extra editable setting
Hint	Highlights a helpful move.	Highlight Duration
Undo	Reverses an eligible recorded move.	Uses Move History Tracker
Shuffle	Rearranges items.	Uses the existing shuffle behavior
Freeze Time	Stops the timer temporarily while allowing play.	Freeze Duration
Extra Time	Adds time to a timed level.	Bonus Seconds
Hammer	Clears one available match.	Include Locked Shelves
Magnet	Attempts to clear up to three available matches.	Include Locked Shelves
Magic Wand	Converts selected item types to a chosen type.	Types To Convert Count
Unlock	Unlocks one shelf, or clears one slot obstacle if no locked shelf is available.	Uses the existing unlock behavior

A booster must have a valid situation to act on. For example, Undo needs a recorded move and Extra Time needs a timed level.

10.2 Change a booster's settings

Select its data asset. The example below shows the **Hammer asset** and the common settings used by boosters.

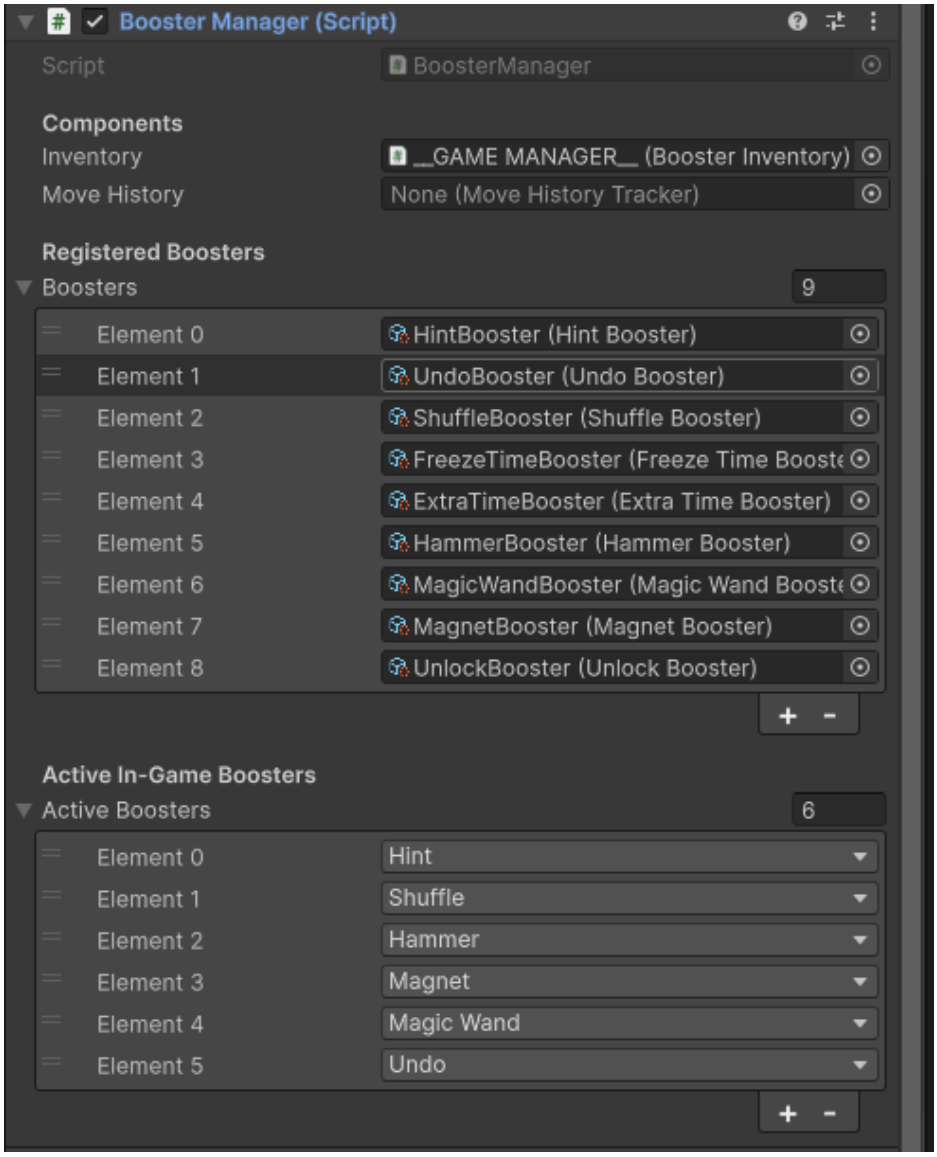


Hammer booster asset and common booster settings

Field	What it does
Booster Type	Connects the asset to its existing booster identity. Keep it matched to the asset's behavior.
Booster Title	Name shown by the relevant booster UI.
Description	Explains the booster to the player.
Icon	Booster artwork.
Unlock At Level	First level where the booster becomes available.
Show Unlock Popup	Enables the introduction popup at its unlock level.
Default Inventory Count	Stored on the asset. In this project's current inventory loading flow, starting stock comes from Booster Inventory's Initial Count Per Booster.
Blocks Player Interaction	Controls input blocking during the booster's action where supported. Freeze Time keeps player interaction available.

10.3 Choose the active boosters

Select the game manager and expand **Booster Manager**.



Registered and active booster lists

- **Inventory:** Reference to Booster Inventory.
- **Move History:** Reference to the tracker used by Undo. Assign it when using Undo.
- **Boosters:** The registered booster assets.
- **Active Boosters:** The booster types offered through the active in-game booster flow.

Keep the required asset registered, then choose the types you want active. Check the booster bar and pre-level selection separately; they also have their own UI references.

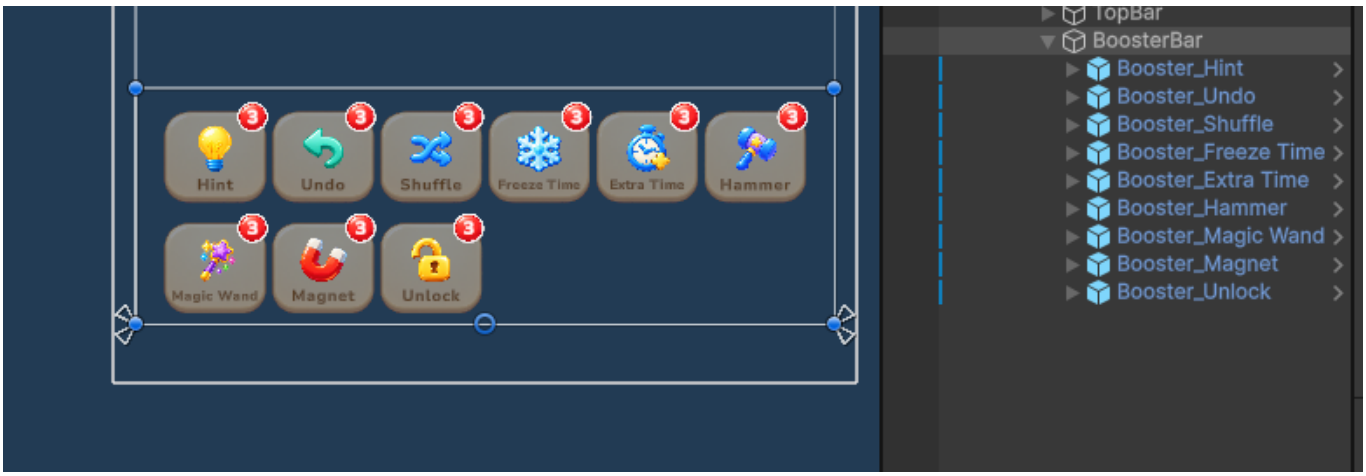
10.4 Change starting stock

On the same game manager, expand **Booster Inventory**.

- **Initial Count Per Booster:** Starting count for each type that has no saved inventory yet.
- **Infinite Boosters:** Stops consumption for testing.

Saved inventory takes priority over new starting values. Turn Infinite Boosters off before your final check.

10.5 Change the booster bar

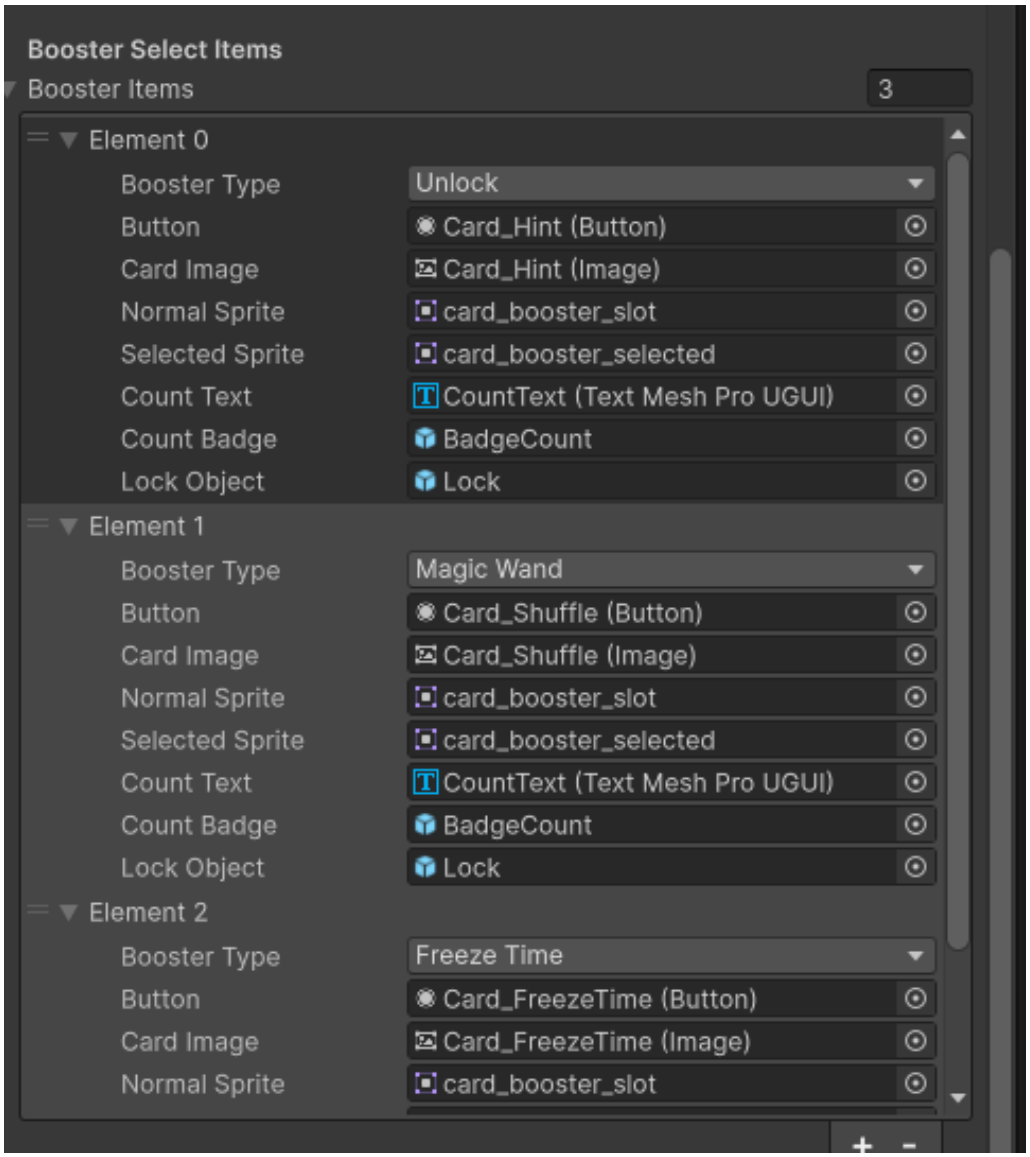


Booster bar objects

Select the relevant button under BoosterBar. Its **Booster Button UI** component has Booster Type and references for the button, icon, name, count, badge, and lock visual.

When duplicating a booster button, set the correct Booster Type and connect its own child objects. Check the **Booster Bar UI > Buttons** list too.

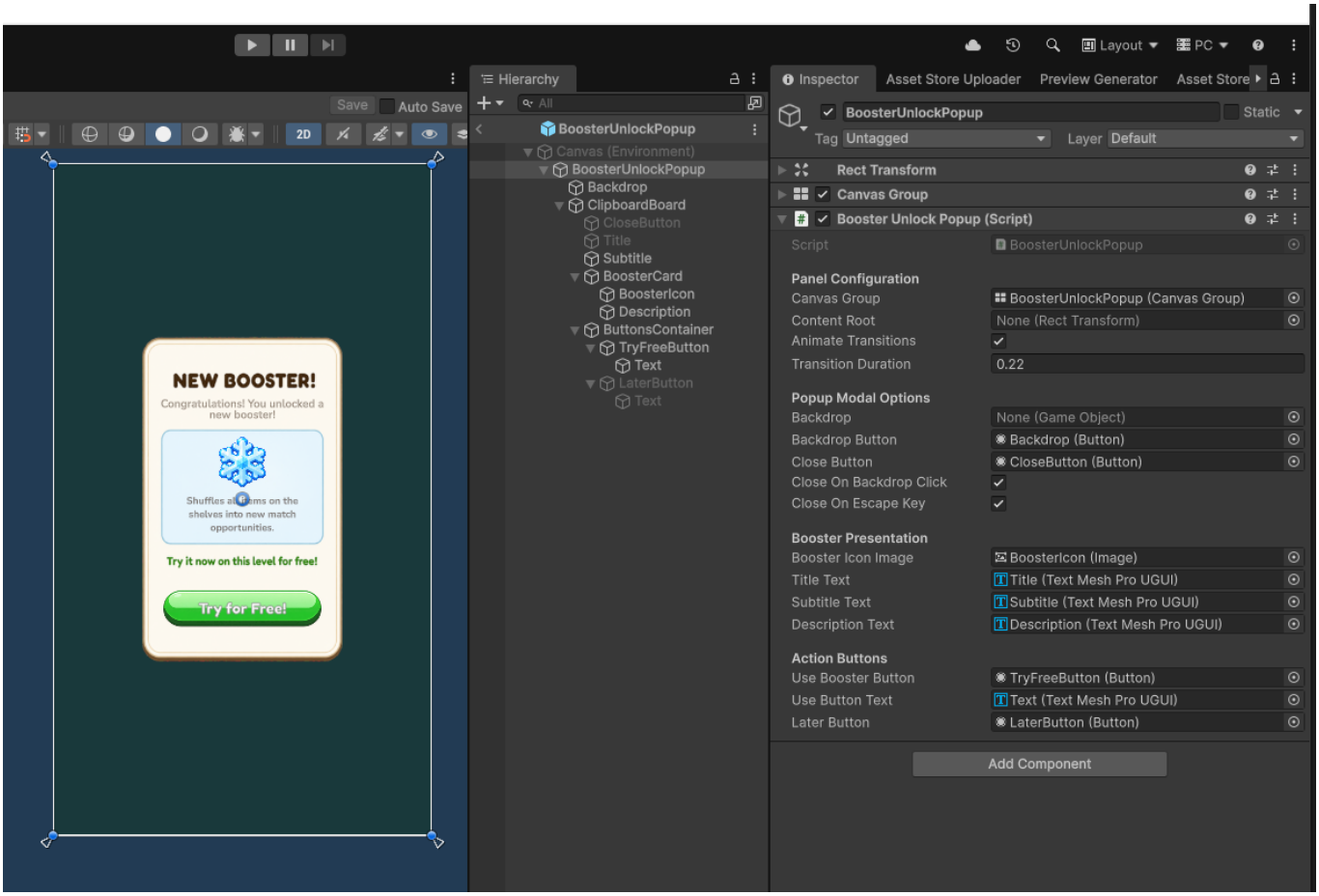
10.6 Change pre-level selection



Pre-level booster cards and their references

In LevelStartPopup, each Booster Select Item connects a **Booster Type**, Button, Card Image, Normal Sprite, Selected Sprite, Count Text, Count Badge, and Lock Object. Change the type to choose the existing booster offered by that card. Keep its child references assigned.

10.7 Change the unlock popup



Booster unlock popup

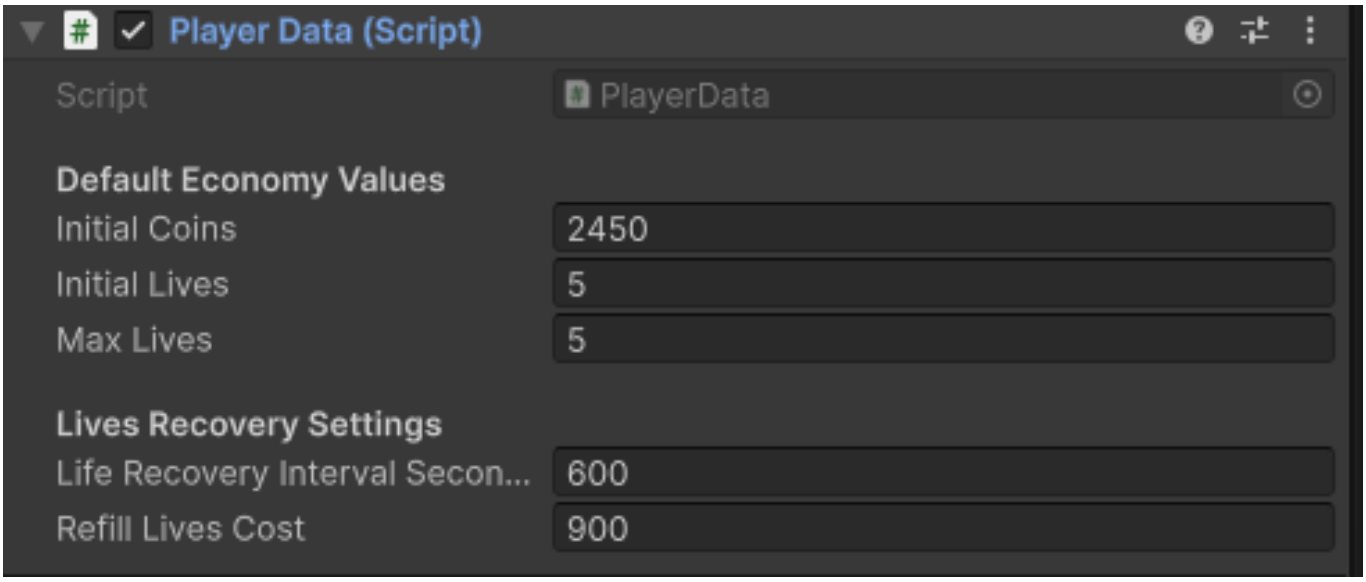
Open BoosterUnlockPopup.prefab. Replace its graphics and adjust the connected title, subtitle, description, icon, and action button. The selected booster supplies its content when opened.

Unlock popup history is saved. A popup already shown on your test profile may not appear again simply because you change its artwork.

11. Change coins, lives, and rewards

11.1 Coins and lives

Select the game manager and expand **Player Data**.



Player Data settings

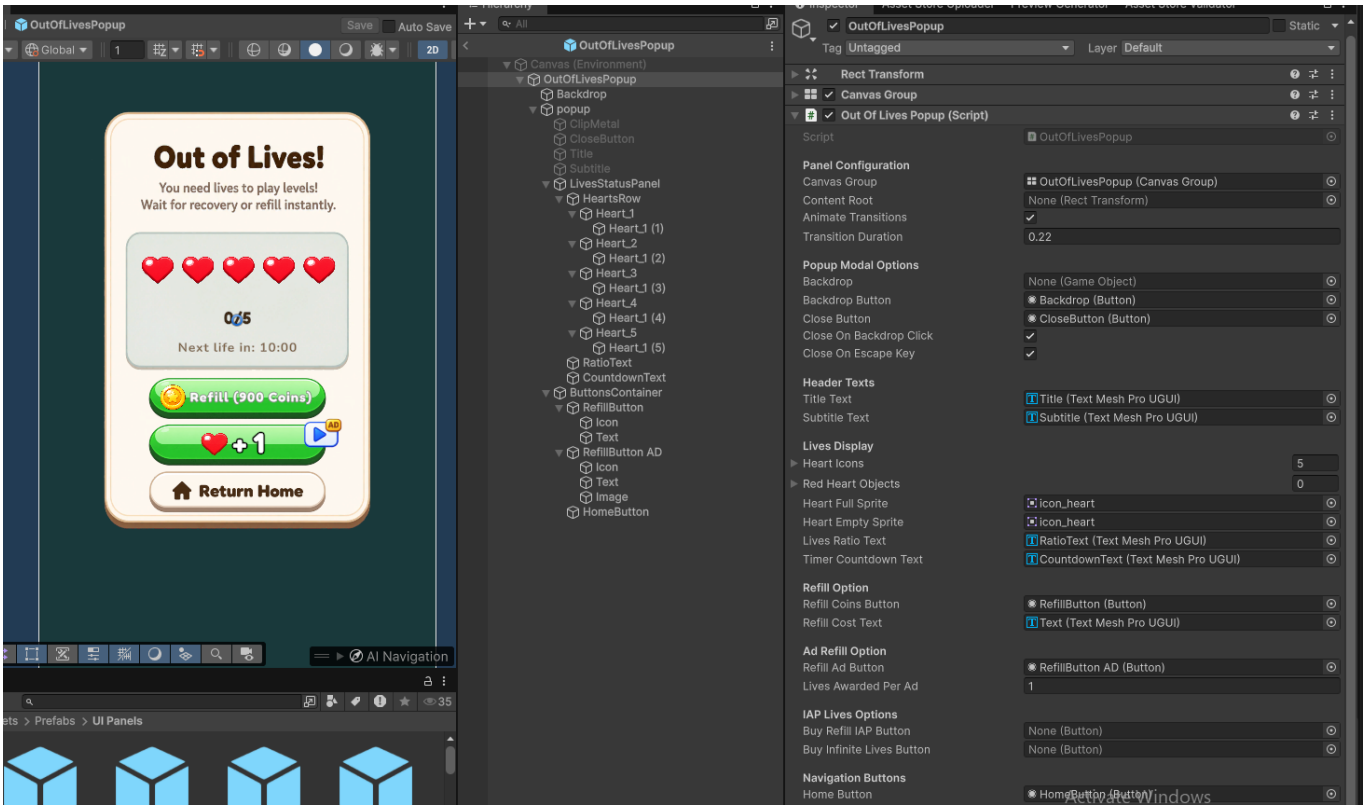
Setting	What it does
Initial Coins	Coins given when no saved coin value exists.
Initial Lives	Starting lives for a new profile.
Max Lives	Normal maximum lives. Keep starting lives at or below this.
Life Recovery Interval Seconds	Time to recover one life. 600 means 10 minutes. Use a positive value.
Refill Lives Cost	Coins needed for a full refill.

Lives recover using saved time, including time away from the game. Infinite lives lets the player attempt levels without consuming a life for the configured duration.

Life use in the current flow: Starting a level does not immediately remove a life. Leaving or restarting an unfinished attempt uses one. Giving up after failure uses one. Completing a level keeps the life. A continue resumes the same attempt.

11.2 Out of Lives popup

Open Assets/Prefabs/UI Panels/OutOfLivesPopup.prefab.



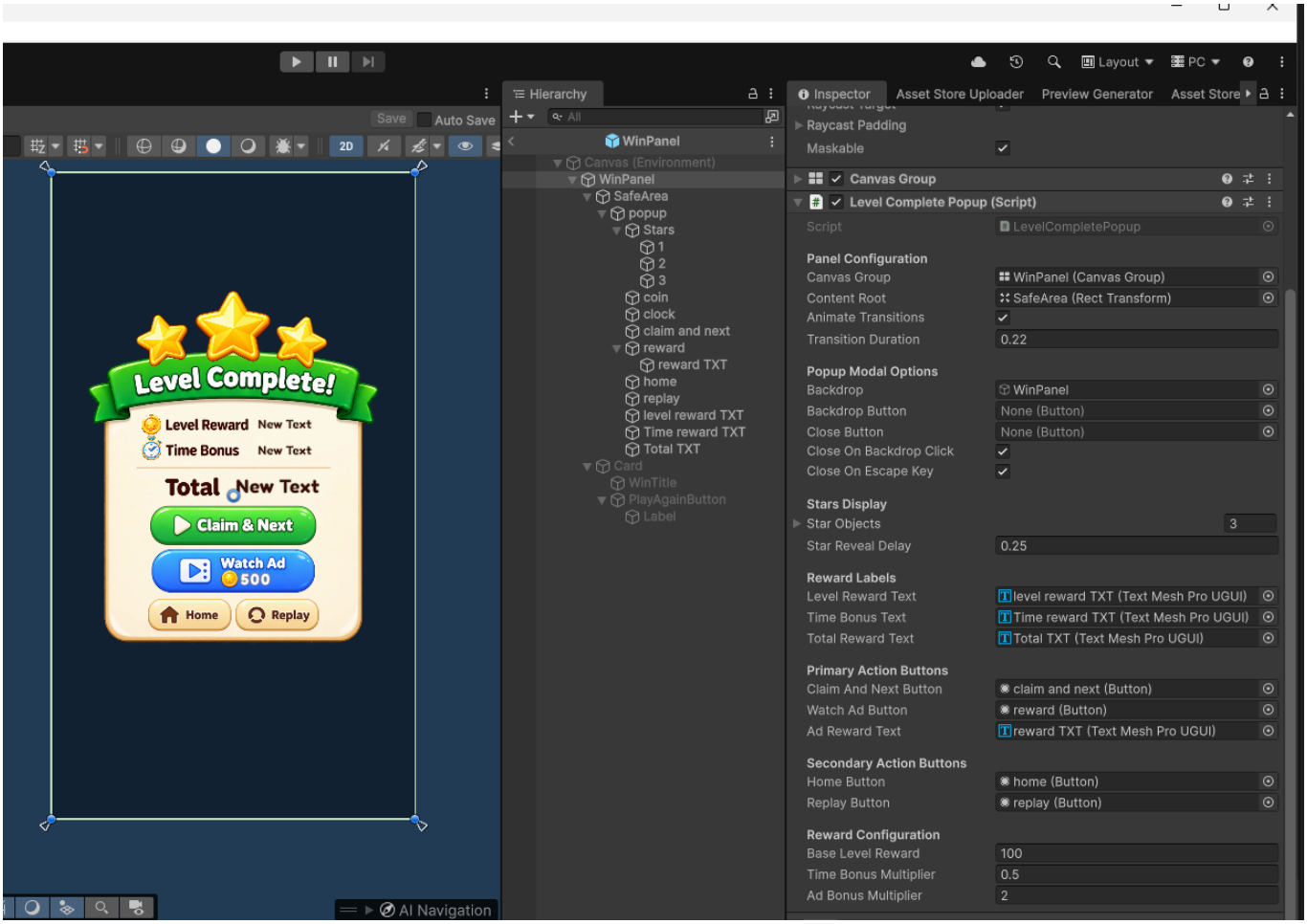
Out of Lives popup and editable references

Area	What you can change
Header Texts	Assign the title and subtitle text objects. Check both edit-time text and the running popup.
Lives Display	Heart objects, full/empty sprites, the lives ratio, and countdown text.
Refill Option	Coin refill button and cost label. The actual coin cost comes from Player Data.
Ad Refill Option	Rewarded button and Lives Awarded Per Ad.
IAP Lives Options	References to the existing refill and infinite-lives purchase buttons.
Navigation	Home button.

If you change Max Lives, check the number of connected heart objects too. A larger number in Player Data does not add new UI objects by itself.

11.3 Level completion rewards

Open Assets/Prefabs/UI Panels/WinPanel.prefab.



Win popup and reward settings

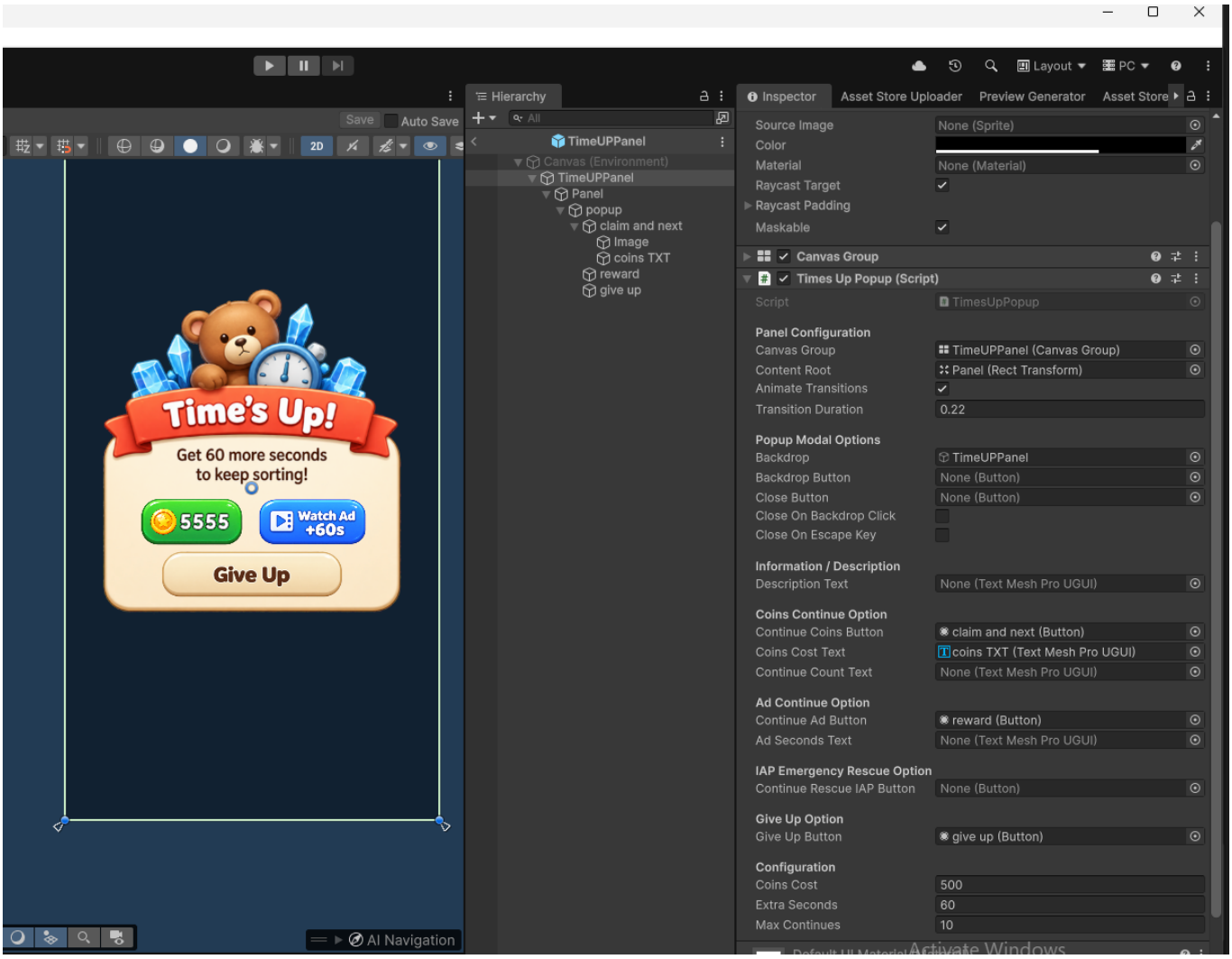
- **Base Level Reward:** The popup's fallback reward.
- **Time Bonus Multiplier:** Controls the reward calculated from remaining time.
- **Ad Bonus Multiplier:** Controls the multiplied claim option.
- **Star Objects / Star Reveal Delay:** Control the star display and reveal timing.
- **Reward Labels and Buttons:** References for displaying and claiming the reward.

Current behavior: The normal Game Flow Controller completion call supplies a base reward of 100. This takes priority over the popup's fallback Base Level Reward. Changing only the popup field will not change that normal base reward. Time Bonus Multiplier and Ad Bonus Multiplier remain useful Inspector settings.

The time bonus is remaining seconds x Time Bonus Multiplier, rounded to a whole number. For example, 40 seconds left with a multiplier of 0.5 adds 20 coins. Stars use the remaining time ratio: at least 50% gives three stars, at least 25% gives two, and less gives one. An unlimited-time level gives three stars. These star thresholds are part of the existing behavior, not Inspector sliders.

11.4 Time's Up and continue

Open Assets/Prefabs/UI Panels/TimeUPPanel.prefab.



Time's Up popup and continue configuration

Field	What it does
Coins Cost	Coins needed for the coin continue option.
Extra Seconds	Time added when continuing.
Max Continues	Limit checked by the continue flow.
Coin / Ad / Rescue buttons	Connect the existing continue choices.
Description and labels	Connect the text objects updated by the popup.
Give Up Button	Ends the attempt through the existing failure flow.

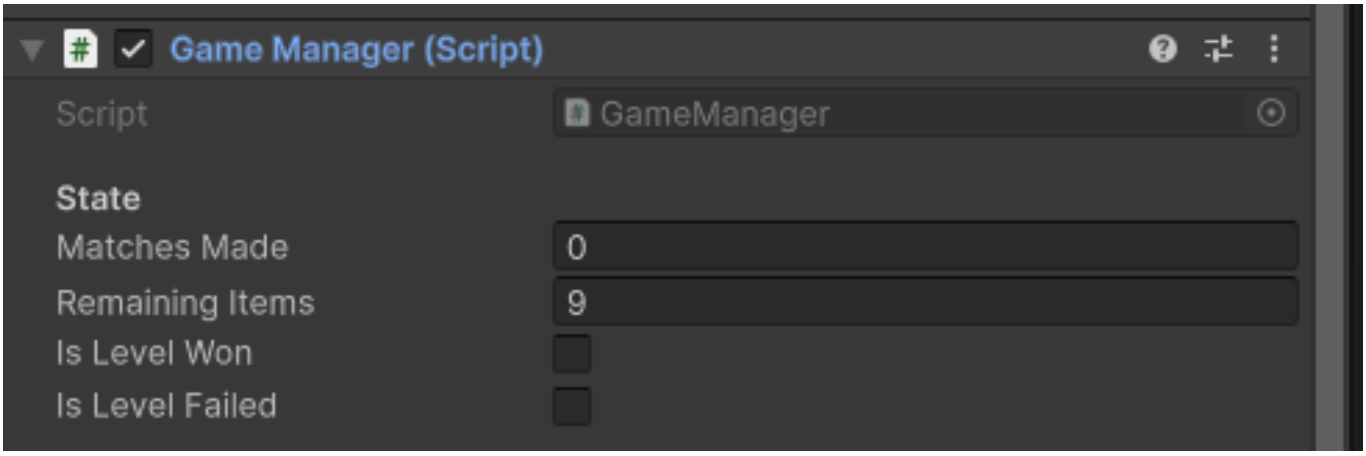
11.5 Progress and testing settings

Use the Level Manager screenshot in Section 6.

- **Level Database:** Main ordered content list.
- **Inline Level Prefabs:** Fallback list when the database has no levels.
- **Level Container:** Optional parent for spawned level content.

- **Save Progression:** Enables saving unlocked progress.
- **Save Key:** Name used for stored progression. Changing it switches to a different stored progression value; it does not rename or clear all player data.
- **Enable Test Level / Test Level Number:** Starts testing from a chosen level.
- **Enable Skip Level Button / Skip Level Button:** Connects a testing shortcut.
- **Legacy Fallback:** Older scene-based setup. Keep it as supplied for the normal prefab/database workflow.

The Game Manager fields below are mostly live status information. They are not the place to set a level's item count or configure its starting difficulty.



Game Manager live status fields

11.6 Quickly test lives options

In Play Mode, open Player Data's component menu. The included helpers are **Debug: Set Lives To 5 (Full)**, **Debug: Set Lives To 0**, and **Debug: Fast-Forward 10 Minutes**. Use these to check the hearts, recovery, and Out of Lives flow. The full-lives helper uses your configured Max Lives, and the fast-forward helper advances one configured recovery interval, even though their menu labels use the original defaults.

These helpers change saved test values. Keep them for your test profile.

12. Understand and adjust the game flow

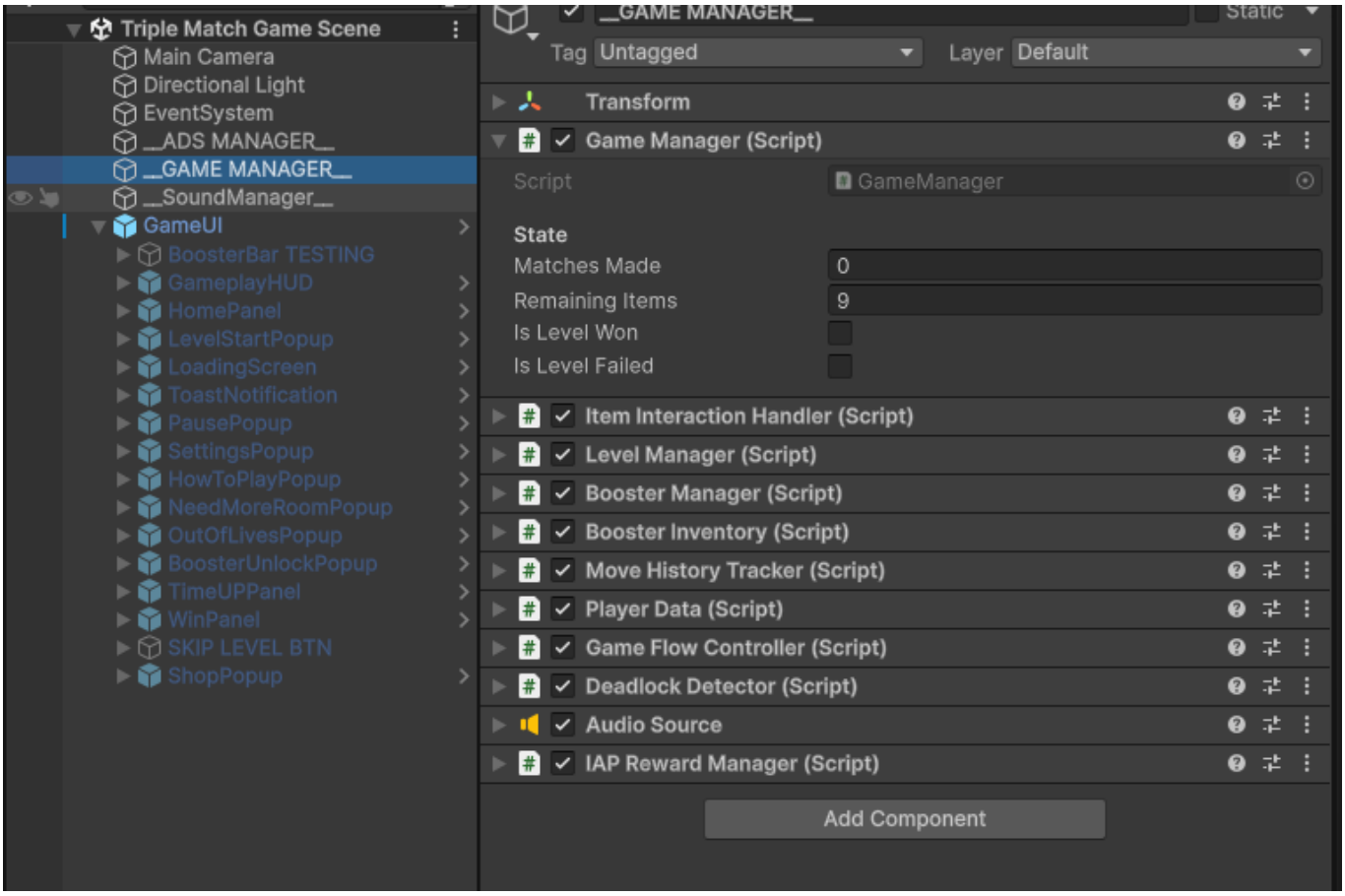
12.1 The basic flow

Loading > Home > Level Start > Gameplay

From Gameplay:

- **Pause > Resume**, restart, or return home.
- **Complete all matches > Level Complete > Claim > Next Level or Home.**
- **Timer reaches zero > Time's Up > Continue or Give Up.**
- **No useful room/moves > Need More Room > Rescue, restart, or home.**
- **No lives before starting > Out of Lives > Wait, refill, or home.**

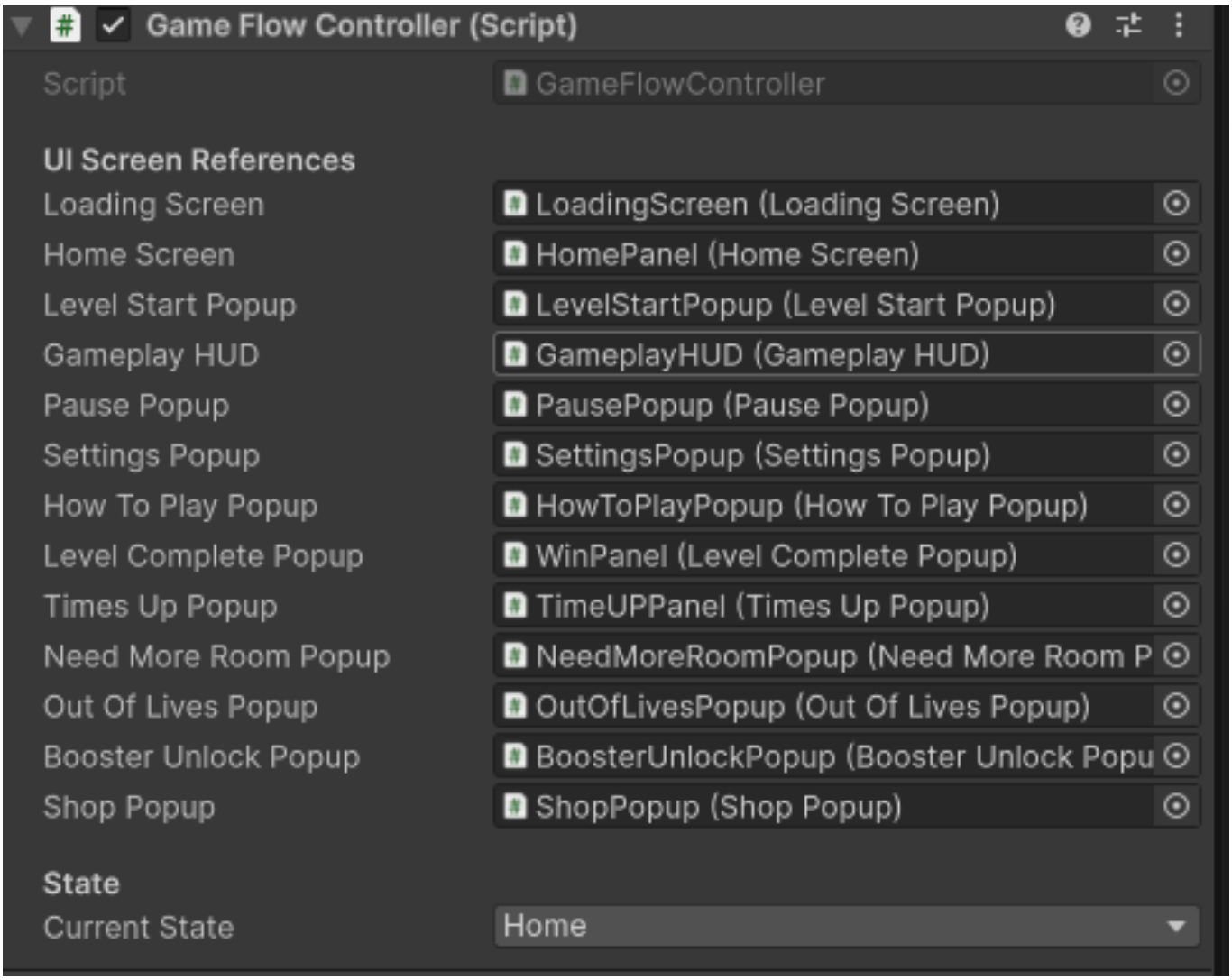
12.2 Which objects control it?



Game managers and UI objects in the scene

Component	Simple job
Game Flow Controller	Moves the player between screens and gameplay states.
Level Manager	Loads levels, manages progression, and runs the timer.
Game Manager	Tracks completed matches and remaining items.
UI Manager	Manages the screen and popup display.
Item Interaction Handler	Handles item input and dragging.
Deadlock Detector	Checks when the board needs a rescue option.
Player Data	Saves coins, lives, and recovery times.
Booster Manager / Inventory	Runs boosters and tracks their stock.

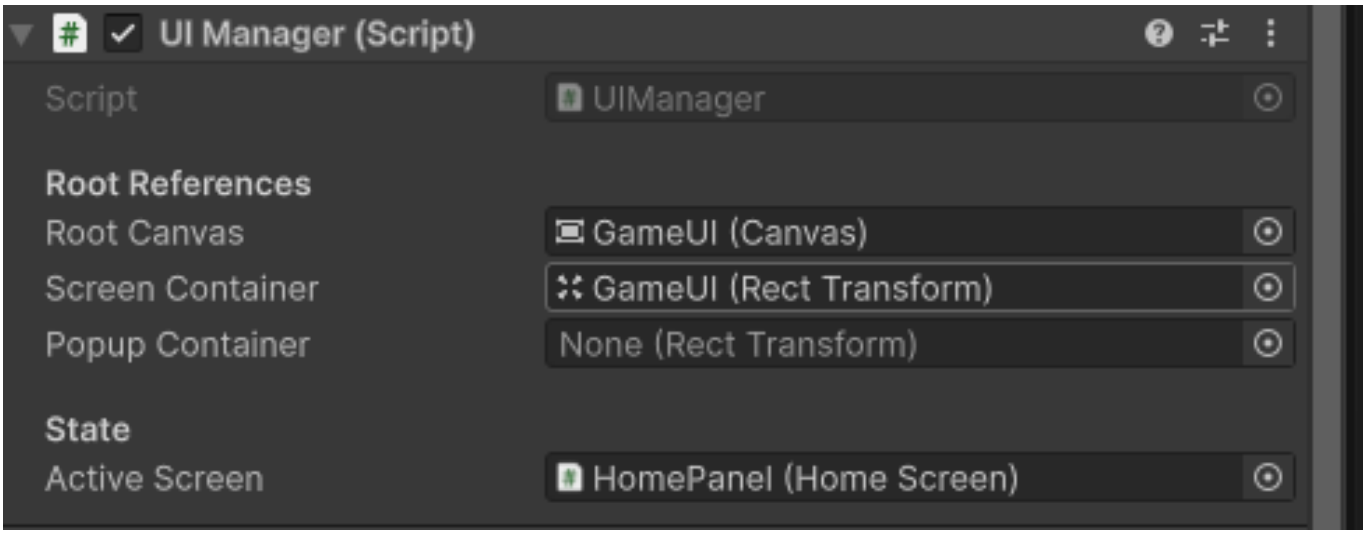
12.3 Replace a screen or popup reference



Game Flow Controller screen references

1. Duplicate the existing screen or popup prefab as your starting point.
2. Change its graphics and layout. Keep its required component and child references.
3. Add the replacement to the scene's GameUI setup.
4. Assign the replacement's matching component in Game Flow Controller.
5. Remove or disable the old scene copy once references have been checked.
6. Test every way of opening and closing it.

Keep one intended active version of each screen. Some components search for their references if fields are empty, so clear assignments make the setup easier to understand.



UI Manager canvas and container references

UI Manager connects the root canvas and screen/popup containers. **Active Screen** shows the current screen. It is not a sequence editor.

12.4 Connect another button to an existing action

For example, to add a second Settings button:

1. Add or duplicate a UI Button.
2. Select it and find **Button > On Click()**.
3. Add an event with **+**.
4. Drag the scene object containing Game Flow Controller into the object field.
5. Choose **GameFlowController > OpenSettings()**.
6. Save the scene and test the button.

Other existing no-argument actions include **OpenShop()**, **OpenHowToPlay()**, **OpenPauseMenu()**, **ResumeGameplay()**, **RestartLevel()**, and **ReturnToHome()**. Use the appropriate existing action for the button's purpose.

Many supplied popup buttons are already connected by their components when the game starts. An empty On Click list does not always mean a button is unconnected. Avoid adding a second call to a supplied button that already works.

12.5 Change popup behavior and timing

On a popup root, check **Animate Transitions**, **Transition Duration**, **Close On Backdrop Click**, and **Close On Escape Key**. Keep the background, close button, and content references connected.

Some gameplay popups force their close rules when opened. For example, Need More Room prevents dismissing the rescue decision through the backdrop or Escape. Test the running result after editing a close option.

These settings let you adjust the existing flow. A completely new game state or a new feature needs a separate implementation.

12.6 Edit a guided tutorial level

Open a level prefab that uses **Tutorial Level Controller**, or add that existing component to a suitable manual level. Use the level's own item and slot children for the references.

Setting	What it does
Steps	Ordered list of guided moves.
Step Name	A label to help you recognize a step in the Inspector.
Source Item	The item allowed to move in that step.
Target Slot	Where the player must place it.
Fallback shelf / slot indexes	Alternative ways to find the source and target. Index 0 means the first entry. Direct references are easier for a first tutorial.
Require Match To Advance	Waits for a match instead of advancing as soon as the item lands.
Hint Text	Stores the step hint; display depends on the existing hint output.
Hand Pointer / Custom Hand Sprite	Uses the existing pointer or your replacement image.
Hand Offset	Moves the hand graphic relative to the guided positions.
Shift To Normal After Complete All Steps	Restores normal play after the last guided step.
Pause Timer During Tutorial	Pauses the timer while guidance is active.
Initial Delay	Wait before starting the first step.

When you move or remove tutorial items, update the step references too. Test all steps from the beginning after editing the layout. Use the level prefab workflow in Sections 5-7; no additional screenshots are needed for this reference table.

12.7 Adjust how the level fits the camera

Select **Main Camera** and find **Camera Grid Fitter**. Top Padding World, Bottom Padding World, and Side Padding World reserve space around the shelves. Horizontal Pixel Margin adds a horizontal screen margin. Min Ortho Size and Max Ortho Size limit the camera's zoom range. Manual Center Offset shifts the fitted center. Smooth Transition and Transition Duration control the adjustment animation.

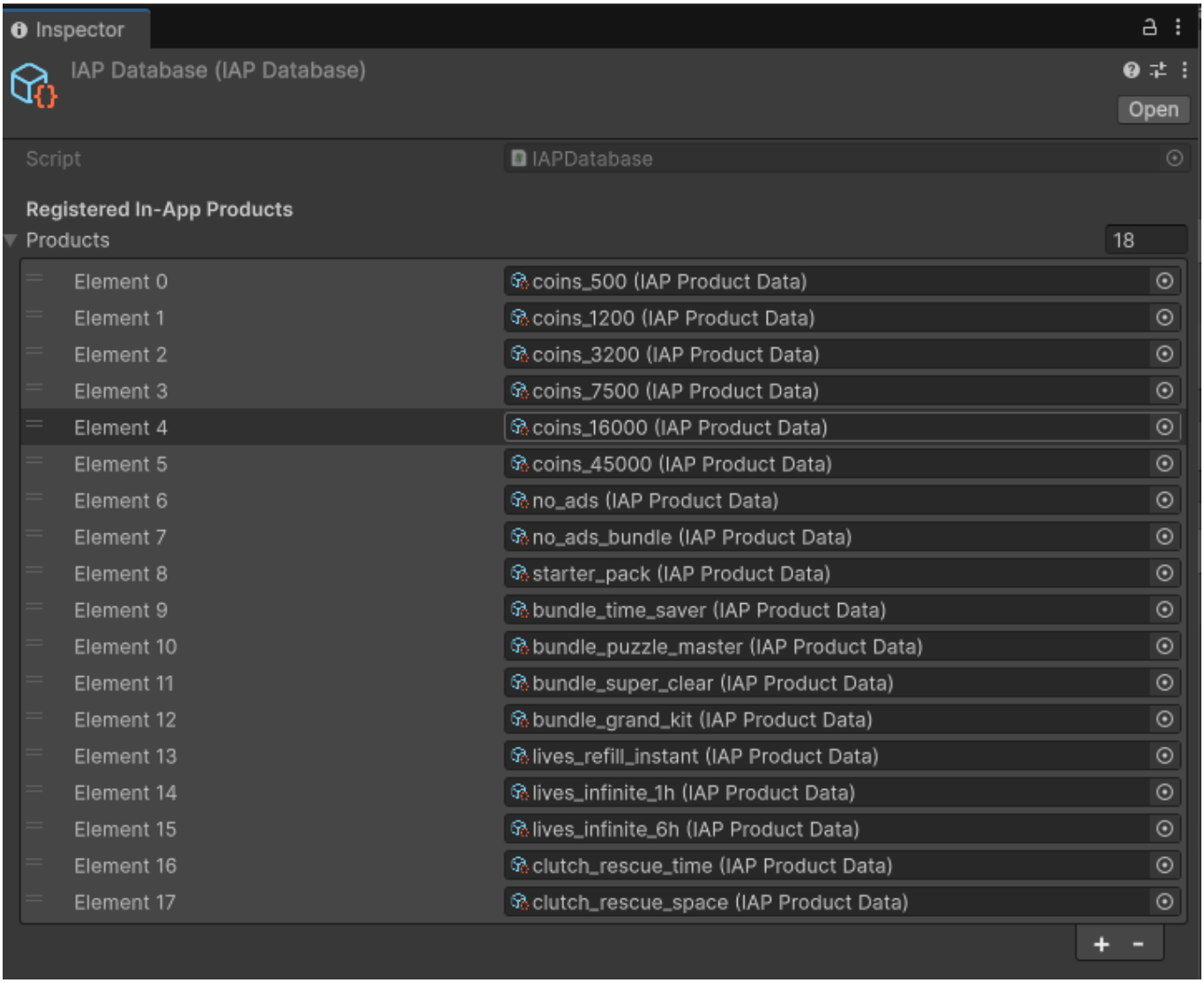
Use these settings when new shelf artwork or a larger grid needs more room. Test small and large levels, and leave room for the top HUD and bottom booster bar. The main scene screenshot in Section 1 shows where to find Main Camera.

13. Change shop products and rewards

13.1 The game-side assets

- **IAP Product Data:** Describes one product and its rewards.
- **IAP Database:** Lists the product assets available to the game's reward lookup.
- **IAP Reward Manager:** Connects successful purchases to the matching product's rewards.
- **Shop Popup:** Displays the shop categories and connects its existing buttons.

Open Assets/In-apps/IAPDatabase.asset to see the registered game-side products.



Game-side IAP Database and product references

Select a product reference, or open an asset in Assets/In-apps/Products, to edit the actual reward data. The separate store catalog is not the same as these game-side product assets.

13.2 Product fields

Field	What it does
Product Key	Unique key linking the purchase to this game's reward definition. It must match the registered purchase product.
Display Name	Product title available to the game UI and delivery message.
Description	Description of the offer.
Fallback Price	A display fallback. Changing it does not set the real store price.
Icon	Product artwork available to UI that uses this field.
Is One Time Only	Records a purchased flag after delivery. The shop must use that state to hide or block the offer; the checkbox alone is not a universal purchase block.
Rewards	The list of rewards delivered for this product.

Existing shop cards can have their own artwork and text. Check and update the visible card too; product metadata is not a promise that every existing card updates automatically.

13.3 Reward choices

Rewards are entries inside a product asset. They are not separate ScriptableObject files in this project.

Reward Type	Fields to set	Result
Coins	Amount	Adds coins.
Booster	Booster Type and Amount	Adds stock for an existing booster.
Infinite Lives	Duration Hours	Grants infinite lives for that duration.
Full Refill Lives	Type	Fills lives to the normal maximum.
Remove Ads	Type	Enables the existing no-ads state.
Unlock Shelf Slot	Type	Runs the existing Unlock booster for free. It needs an eligible lock or obstacle during gameplay.

Only fields relevant to the selected reward type are used. For example, Duration Hours is used for Infinite Lives, while Amount is used for coins and boosters.

13.4 Make a new reward pack without changing scripts

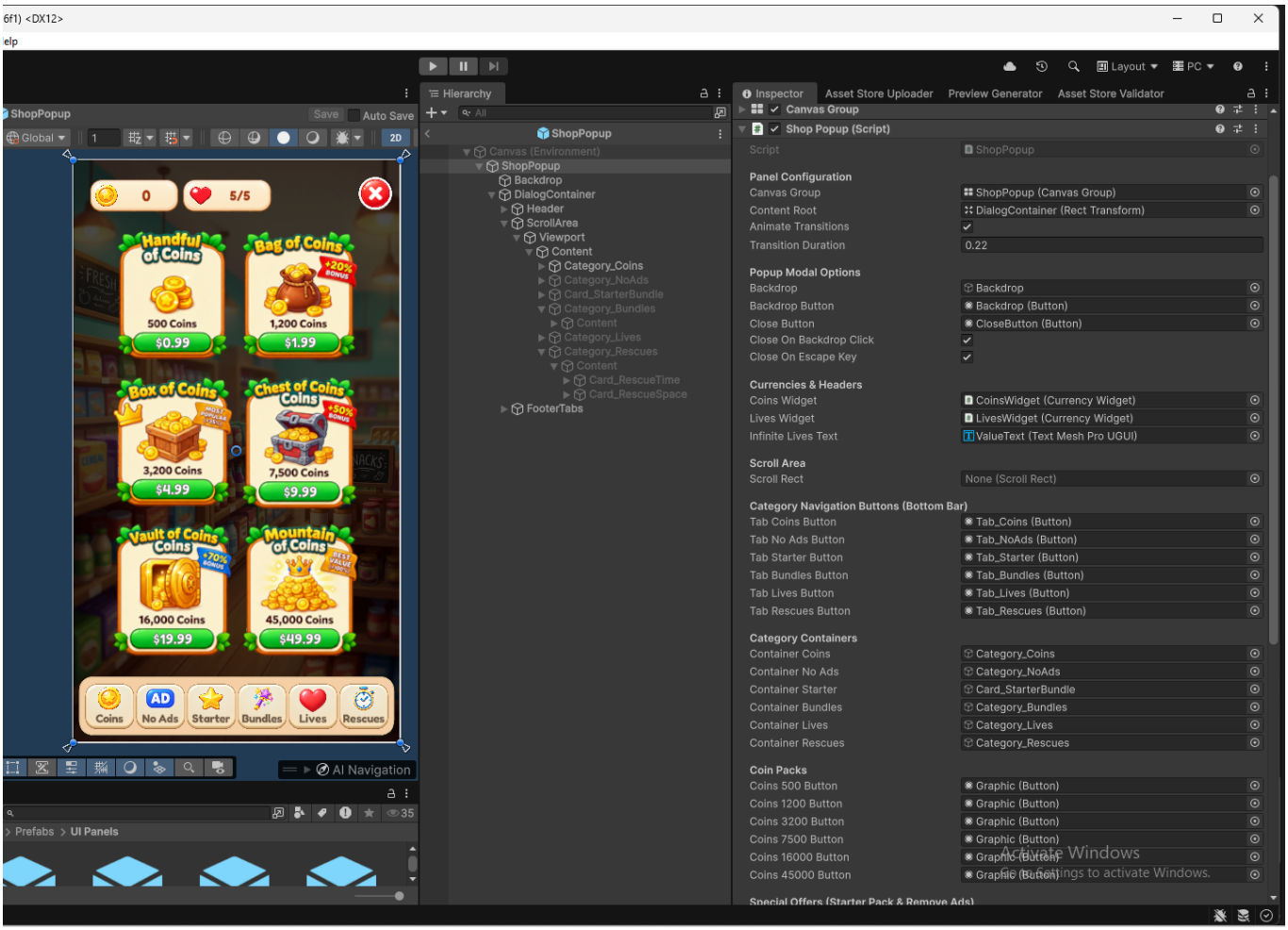
1. Open Assets/In-apps/Products.
2. Duplicate a similar product asset using the Project window.
3. Rename the new asset.
4. Give it a unique Product Key.
5. Set its title, description, artwork, and display fallback.
6. Edit the Rewards list.

7. Add the new product asset to IAPDatabase > Products.
8. Ensure the game manager's IAP Reward Manager uses this database.
9. Connect a purchase button to the matching registered product through the existing purchase-button integration.
10. Test the purchase result and all rewards.

Example: A Weekend Pack can contain Coins > 2000, Booster > Hammer > 3, and Infinite Lives > 2 hours. These existing reward types can be combined without adding a new reward script.

Creating the reward asset does not register a real store product. Its key also needs to exist in the purchase setup. Store and SDK setup are covered separately.

13.5 Shop layout and new calling



Shop categories, containers, and button references

The supplied shop has categories for Coins, No Ads, Starter, Bundles, Lives, and Rescues. Its Inspector connects the category buttons, content containers, and existing purchase buttons.

For an existing offer, change its game-side reward asset and update the card's visible content. Assigning a replacement Button to an existing named field keeps that field's existing product call.

For a **new product key**, do not assign its card to an unrelated old product field: the supplied Shop Popup binds those fields to specific existing keys. Use the included **Bcyly IAP Button** integration on a new card, select its matching registered product, and let that component own the purchase click. Avoid also assigning that button to an old auto-bound purchase field.

The simple purchase flow is:

Purchase button > purchase system > successful purchase > IAP Reward Manager > matching Product Key in IAP Database > Rewards list > coins/lives/boosters update.

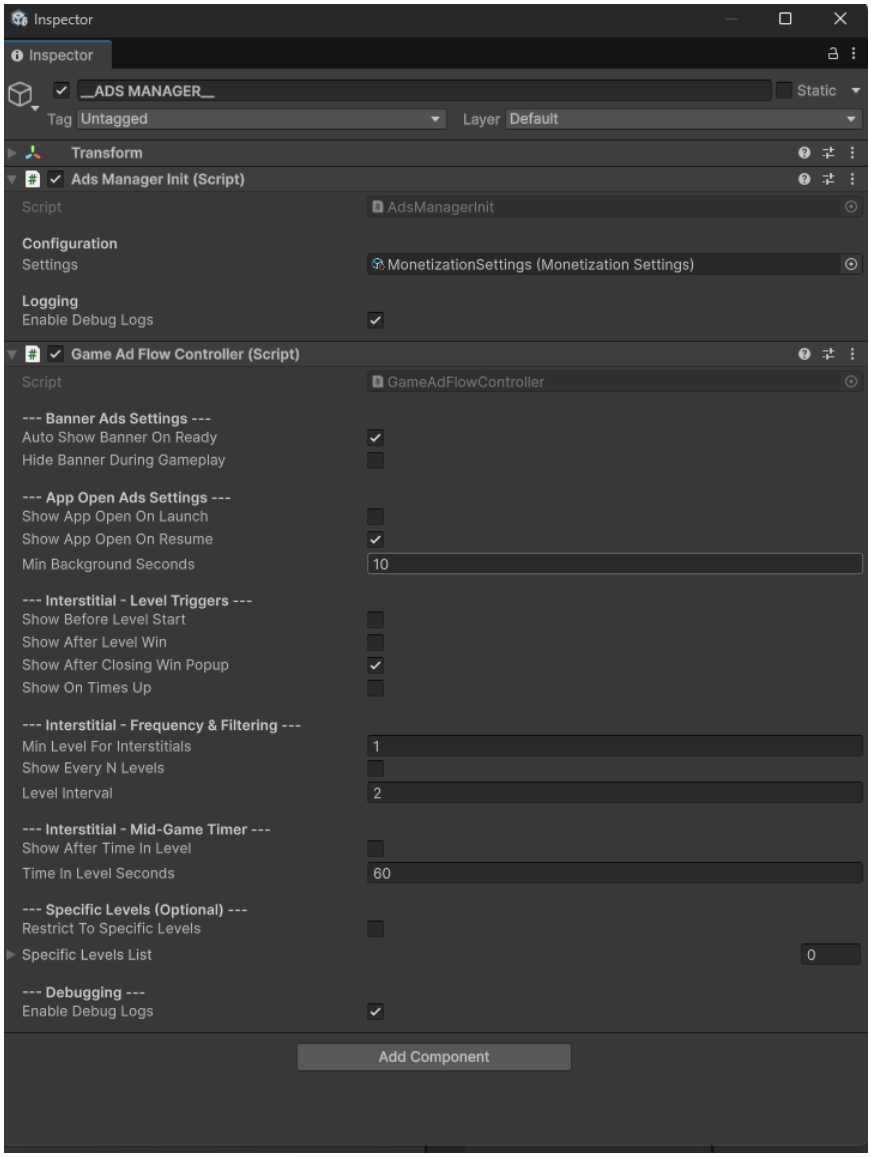
For an extra purchase location, reuse the existing purchase-button integration. Do not connect the product's **Deliver** action directly to a normal purchase click; delivery belongs after successful purchase confirmation.

The existing time and space rescue popups listen for specific rescue product keys. A new general reward pack does not automatically become a new rescue behavior.

14. Choose when ads appear

This section covers game-side timing and calls only.

Select the ads manager object in the scene and expand **Game Ad Flow Controller**.



Game-side ad flow settings

Area	Settings and meaning
Banner	Auto Show Banner On Ready; Hide Banner During Gameplay.
App Open	Show App Open On Launch; Show App Open On Resume; Min Background Seconds before a resume is eligible.
Level triggers	Show Before Level Start; Show After Level Win; Show After Closing Win Popup; Show On Times Up.
Frequency	Min Level For Interstitials; Show Every N Levels; Level Interval.
Mid-game timer	Show After Time In Level; Time In Level Seconds.
Specific levels	Restrict To Specific Levels and its level list.
Debugging	Enable Debug Logs for checking triggers.

These settings decide when the game requests an ad. Availability still depends on the connected ad system.

Existing rewarded calls

Location	What happens after a successful reward result
Level Complete	Claims the multiplied reward.
Time's Up	Adds the configured continue time and resumes.
Out of Lives	Adds Lives Awarded Per Ad.
Need More Room	Attempts the configured rescue booster.

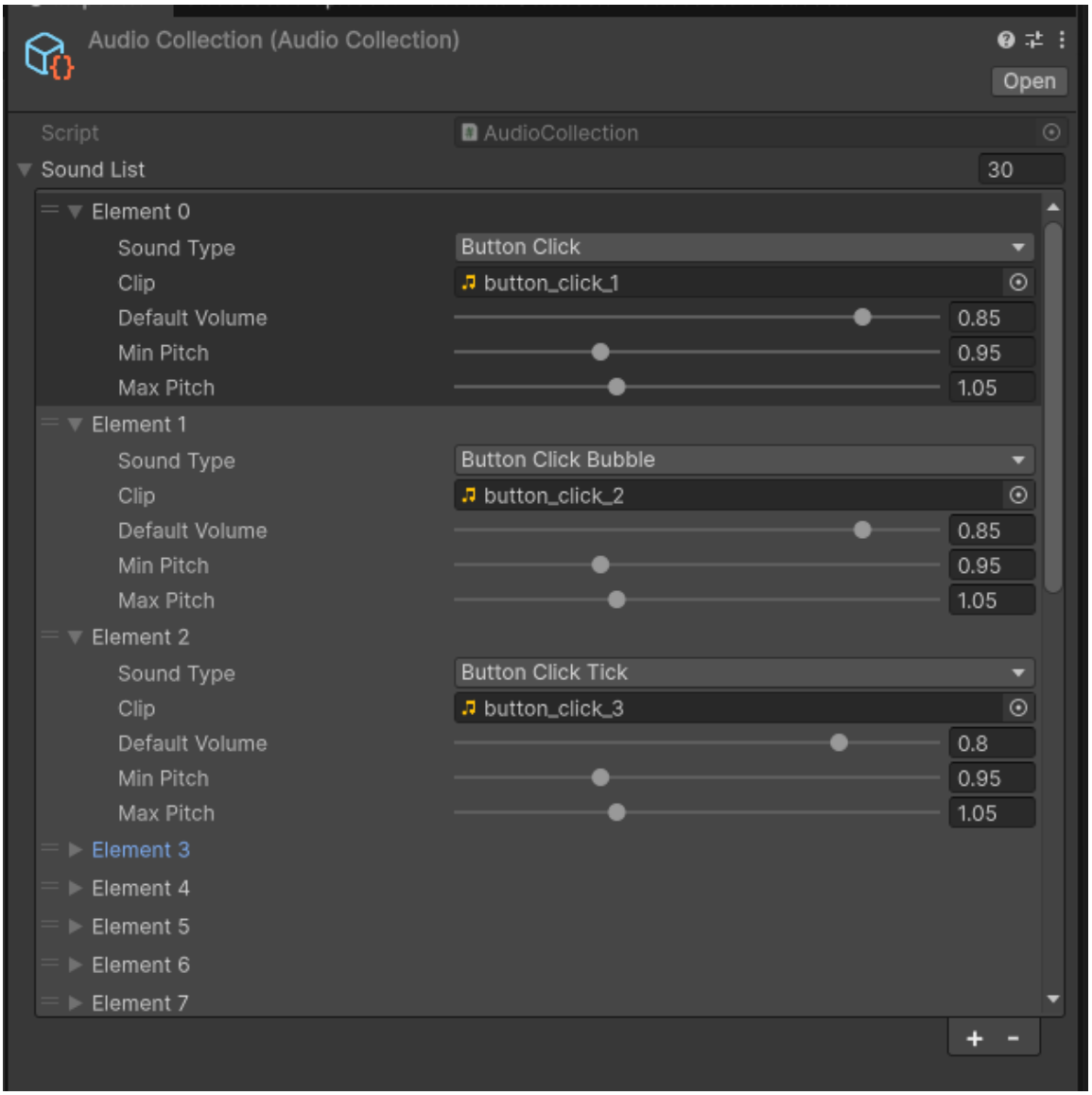
To move an existing rewarded action to another button, replace the Button reference in that popup's component. Keep the popup responsible for the call and reward result.

A second independent rewarded placement needs a result-aware action. The current Game Ad Flow Controller does not provide a generic Inspector button action for choosing any reward. Reuse a supported popup flow for a no-code change. A different reward behavior needs its own integration; simply showing an ad is not enough to safely deliver a reward.

15. Change music and sounds

15.1 Edit the Audio Collection

Open Assets/SFX/AudioCollection.asset.



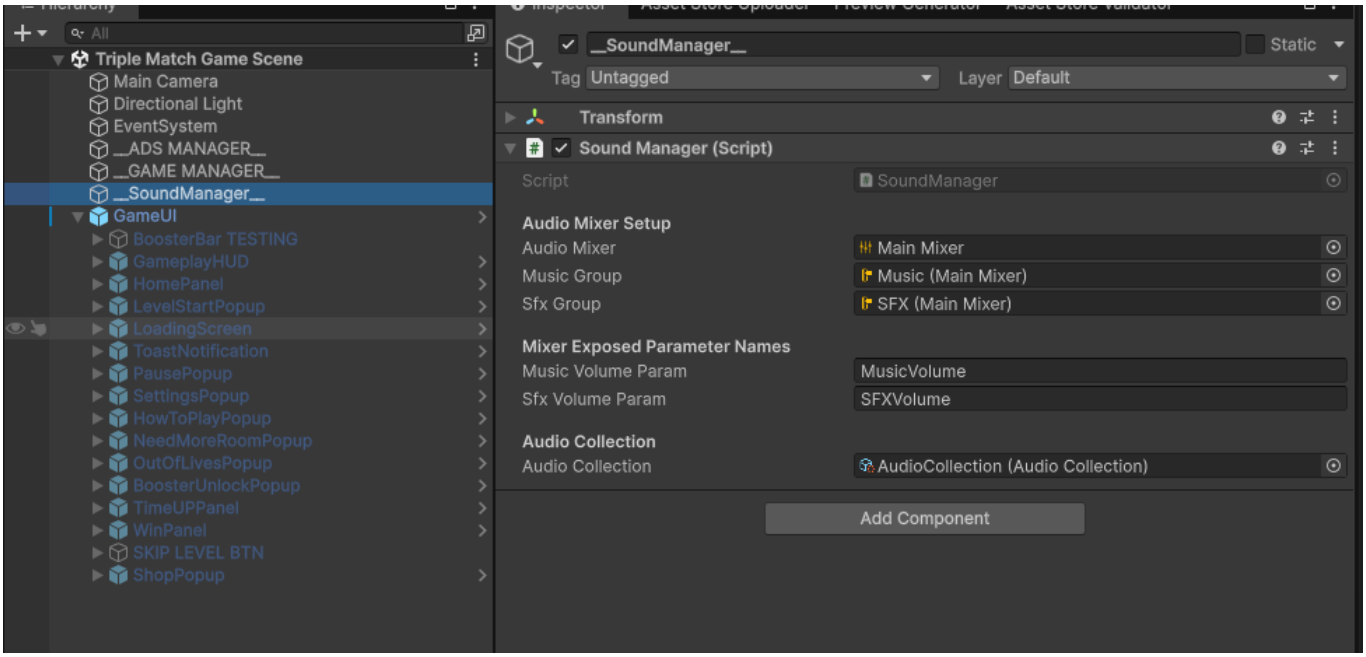
Audio Collection sound entries

Field	What it does
Sound Type	The existing game action or music identity.
Clip	Audio played for that type. Replace this to change the sound.
Default Volume	Starting loudness for the entry.
Min Pitch / Max Pitch	Pitch range used for variation. Use equal values for a fixed pitch.

Replace clips for the existing Sound Types to rebrand the audio without changing scripts. Keep the event types that the game already calls. A newly invented sound event needs a call from a feature.

The current flow uses **Music Sunny Market** on Home and **Music Cozy Breeze** during Gameplay. Replace their clips in the collection to change those tracks.

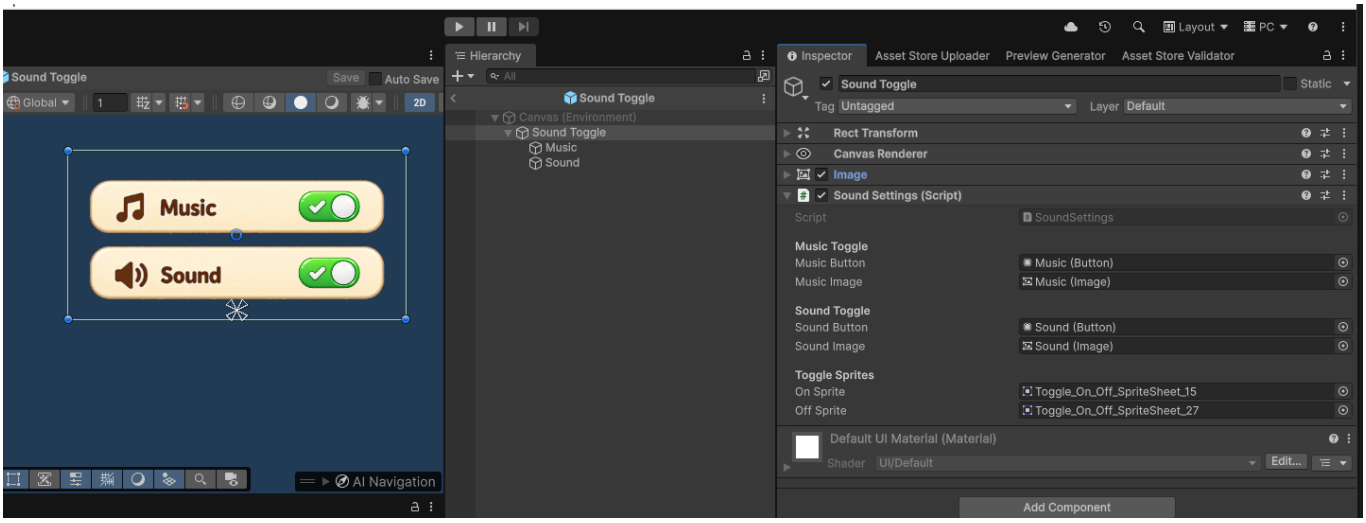
15.2 Check Sound Manager



Sound Manager audio routing and collection

Sound Manager connects the **Audio Mixer**, **Music Group**, **SFX Group**, and **Audio Collection**. Its parameter names must match the exposed mixer parameters. Keep the supplied routing unless you are intentionally changing the audio setup.

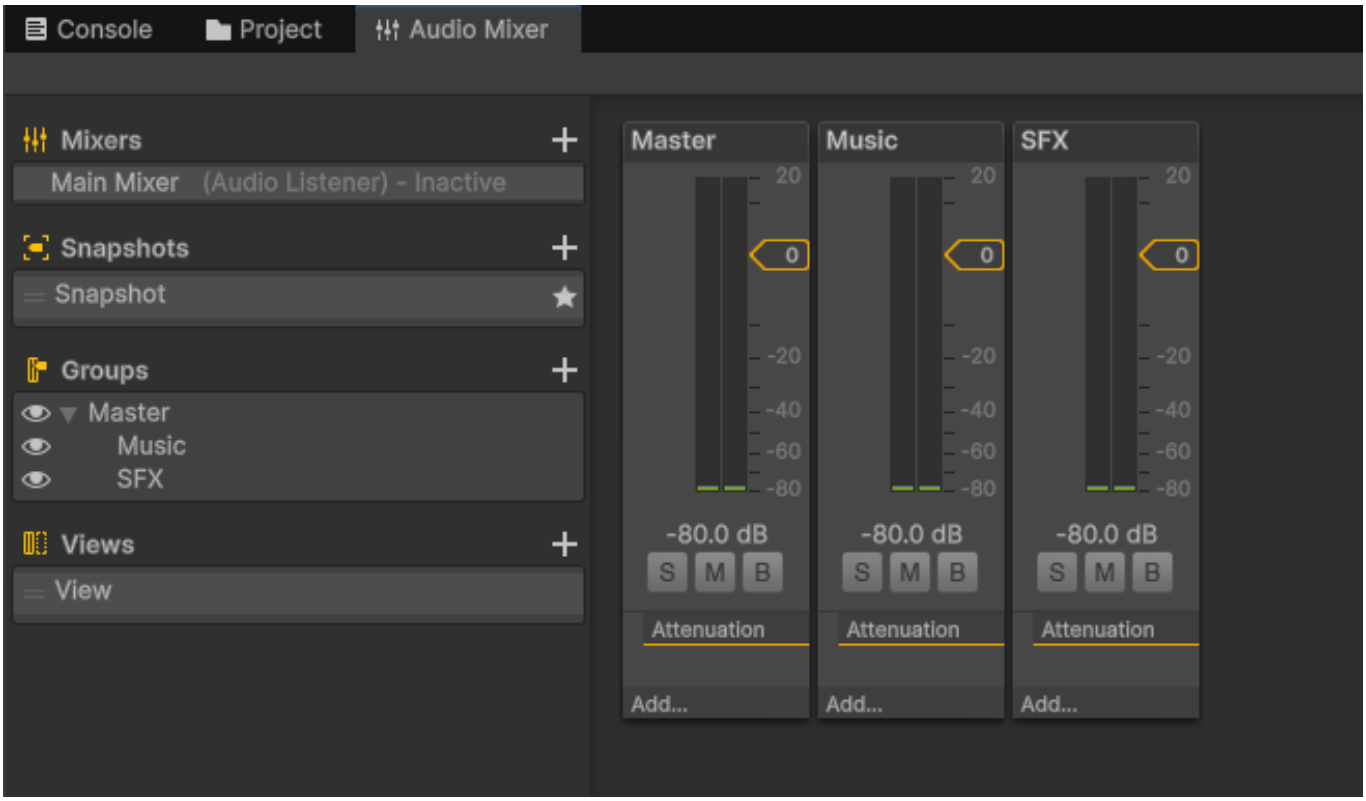
15.3 Change settings toggle graphics



Music and sound toggle UI

The Sound Settings component connects music and sound buttons/images, plus the on/off sprites. Replace the sprites to rebrand the toggles. Keep the button references so the existing actions continue to work.

15.4 Balance music and effects



Main audio mixer groups

Open Assets/SFX/Main Mixer.mixer. Master affects the overall output; Music and SFX separate the two groups. Test with sound enabled in the game's settings because saved mute settings can override what you expect to hear.

16. Important data assets

A data asset is a reusable settings file. Select it in the Project window to edit it in the Inspector.

Asset	Location	Main purpose	Guide section
Item Data	Assets/Data/Items	Item identity, sprite, and name	Items
Level Database	Assets/Data/Levels Database/LevelDatabase.asset	Ordered level prefab references	Levels
Booster assets	Assets/Data/Boosters	Existing booster identity, presentation, unlock, and effect settings	Boosters
IAP Product Data	Assets/In-apps/Products	Product metadata and a list of existing reward types	Shop rewards
IAP Database	Assets/In-apps/IAPDatabase.asset	Product lookup for reward delivery	Shop rewards
Audio Collection	Assets/SFX/AudioCollection.asset	Audio clips and playback defaults for existing sound types	Audio

The linked chapters include the supplied images for these assets. Product rewards are nested list entries. Levels, shelves, items in the scene, and UI panels are prefabs or scene objects rather than ScriptableObjects. Player Data and Booster Inventory are components that also read saved player values.

17. Test your changes

Use this checklist after a content or rebranding pass.

- [ ] Open the main scene and run through Loading, Home, and Level Start.
- [ ] Check the new name, icon, logo, backgrounds, buttons, and fonts.
- [ ] Check the UI at phone and tablet screen sizes.
- [ ] Check new items on the front layer and behind other items.
- [ ] Verify that each manual item count forms complete groups of three.
- [ ] Save and playtest every hand-built level.
- [ ] Check the beginning, middle, and end of a generated batch.
- [ ] Check that the database has the intended order and no missing entries.
- [ ] Test locks, restrictions, movement, and obstacles used by your levels.
- [ ] Test pause, resume, restart, return home, win, failure, and continue.
- [ ] Check booster icons, unlock levels, stock, and pre-level selection.
- [ ] Test lives recovery and Out of Lives options.
- [ ] Test product reward amounts and the visible shop card information.
- [ ] Check the existing ad triggers and rewarded results.
- [ ] Check music, effects, and mute toggles.
- [ ] Turn off test-level mode, skip shortcuts, infinite boosters, and preview overrides you used for testing.
- [ ] Save the scene and all edited prefabs.

18. Common questions

My new item does not appear in the palette.

Make sure an Item Data asset was created under Assets/Data. Check its Icon, then refresh the palette. An imported image alone is not an item definition.

My new item does not appear in an old auto level.

Check that level's saved item pool and selected range. It may still use the old pool. Random selection also does not guarantee every item appears in one run.

Different-looking items are matching as the same type.

Check Item Id values. Each distinct item type needs its own identity. Do not duplicate an Item Data asset and leave its ID unchanged when creating a new type.

My item looks too large, small, or low on the shelf.

Compare the sprite's transparent margins, pivot, and Pixels Per Unit with the old art. Then check ItemPrefab auto-fit settings and shelf slot offsets. Check front and deeper layers.

My manual level cannot be completed.

Check that every item type has a count divisible by three, that the player has room to move, and that locks/restrictions do not block essential items. Run validation and the solver, then playtest.

I painted items, but the level changes when I start it.

Check Distribution Mode. Auto Distribute creates its content from settings. Use Manual for a painted arrangement you want to keep.

My level is missing or the wrong level starts.

Check Save Prefab, the Level Number, the numbered prefab, and the database entry. Click Sync DB if needed. Check Enable Test Level and saved progression too.

I changed the database order, then it changed back.

Sync DB rebuilds order from the numbered prefab files. Keep the numbering and database consistent for a lasting change.

My edits disappear when I stop the game.

You probably changed a running scene object in Play Mode. Stop Play Mode first and edit its saved scene or prefab. Use Save Prefab for the Level Editor's working copy.

I changed starting coins, lives, or booster stock, but the old value remains.

Those starting settings are used when no saved value exists. Test with a fresh profile or an intentional test-data reset. Resetting player data also removes progress and balances, so keep your normal test profile if you need it.

A booster is missing or locked.

Check its registered asset, Active Boosters list, Unlock At Level, stock, button's Booster Type, and pre-level card configuration. The button and asset must refer to the same existing type.

I changed Default Inventory Count on the booster asset, but the starting stock did not change.

The current Booster Inventory loader uses Initial Count Per Booster for unsaved stock. It also restores saved counts. Use that component's starting setting and a fresh test profile.

I changed the shelf prefab, but an old level looks the same.

Check whether that level uses a different shelf variant or has a sprite override. Inspect the actual shelf in the level. Test both an existing level and a newly generated one.

A button has an empty On Click list, but still works.

Many supplied components connect buttons when the game starts. Check the owning component before adding another event, or the action may run twice.

My product asset exists, but no new shop card appeared.

The database registers reward data. It does not automatically build new shop cards. Create or duplicate a card, update its visible content, and connect its matching registered purchase product.

I changed the fallback price, but the real price is different.

Fallback Price is display data. It does not set the store price.

I changed Base Level Reward, but the normal win still gives the same base amount.

The normal game flow supplies 100 when opening the completion popup. That supplied value takes priority over the popup fallback. See Level completion rewards.

Where should I start with a full rebrand?

Change the name and icon, then the loading/home screens, item sprites, shelf graphics, popup artwork, and sounds. After that, create one sample level and check the full game flow. Once the sample looks right, continue with the rest of the content.

Ready to customize: Run the main scene, make one small change, save it, and test it. Use the task table near the start of this guide whenever you need to find the next setting.